Final Report

March 2006

Title:	Conventional Training Versus Game-Based Training
Solicitation Topic Number:	N06-T006
Contract Number:	N00014-06-M-0241
Principal Investigator:	Dr. Tricia Mautone
TPOC:	Curtis Conkey
Period Covered by the Report:	August 1, 2006 – March 2, 2006
Contract Period of Performance:	August 1, 2006 – May 31, 2007
Contractor Line Item Number:	0001AC (Final Report)
Name of the Activity	Office of Naval Research
Contract Expiration Date:	May 31, 2007
Security Classification:	Unclassified
Distribution Statement:	Approved for public release; distribution is unlimited.

Prepared by:
Dr. Tricia Mautone and Dr. Alan Spiker, Anacapa Sciences
Dr. Ron Karp, Arizona State University

Anacapa Sciences, Inc.
301 E. Carrillo Street
Santa Barbara, CA 93101
805-966-6157
tmautone@anacapasciences.com

Report Documentation Page						*Form Approved OMB No. 0704-0188*
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.						
1. REPORT DATE **06 MAR 2007**		2. REPORT TYPE **Final Report**			3. DATES COVERED **01-08-2006 to 02-03-2007**	
4. TITLE AND SUBTITLE **Conventional Training Versus Game-Based Training**					5a. CONTRACT NUMBER **N00014-06-M-0241**	
					5b. GRANT NUMBER	
					5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) **Patricia Mautone; Alan Spiker; Ron (AZ State University) Karp**					5d. PROJECT NUMBER	
					5e. TASK NUMBER	
					5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Anacapa Sciences, Inc.,301 E. Carrillo St. FL 2,Santa Barbara,CA,93101**					8. PERFORMING ORGANIZATION REPORT NUMBER **1573**	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **Office of Naval Research, NAVAIR - Training Systems Division, Curtis Conkey, 12350 Research Parkway, Orlando, FL, 32826**					10. SPONSOR/MONITOR’S ACRONYM(S)	
					11. SPONSOR/MONITOR’S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT **Game-based training is a promising medium for training effectiveness, particularly in military training environments; however, there are few empirically-supported guidelines on how to best implement this approach. The development of a theoretically-driven, empirically-supported tool specifying when and how specific gaming elements can be effectively integrated into training would allow instructional designers in many domains to make informed decisions about how to use game-based training to enhance learning and performance. This report describes how we began developing and evaluating such a tool. We created a taxonomy of serious game elements and integrated it with a taxonomy of cognitive functions derived from a task analysis of a flight training environment. This game element-training competency crosswalk forms the foundation of the proposed tool. We applied principles outlined in the crosswalk to develop specifications for a real-world training game for a flight training program, which will then be used to test premises and tool feasibility in controlled studies. The report details how the findings and products can be applied to other training domains.**						
15. SUBJECT TERMS **15. SUBJECT TERMS Game-based training, Flight management system, Serious games, Callout procedures, training effectiveness**						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **79**	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**				

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Overview of Phase I Research: Objectives and Approach

The primary technical objective of the Phase I research was to develop a plan for the design and testing of a theoretically- and empirically-based tool to aid instructional designers and others in determining, 1) when the integration of gaming elements into instruction is appropriate (i.e., when it is likely to enhance desired training outcomes), and 2) which specific types of game elements would optimize specific training objective outcomes – particularly training objectives commonly found in Navy courses. We refer to this tool as TARGET – the **T**ool for **A**pplying **R**obust **G**aming **E**lements to **T**raining. The tool will, in essence, link specific training objectives and environments with specific game elements, providing output about whether game-based training is a good candidate for the particular training situation, and offering recommendations for optimal game element variations to embed in instruction. The tool will be structured as an XML relational database that is populated with the results from our own empirical research and with "best practices" information from other sources. Its XML construction will permit efficient collection of the most recent findings using the Web, where sources such as the Serious Game ListServe and others like it will be an effective conduit for new information.

Several technical objectives supported the primary goal. The first (Task 1) was to conduct a training needs analysis of the pilot development program curriculum at Arizona State University (ASU)'s in order to develop a taxonomy of the cognitive functions that underlie the training objectives of two core competencies from ASU-AMT's student-pilot training: (1) programming the flight management system (FMS) and (2) executing verbal callout procedures. The training objectives at ASU-AMT are similar to those found in many Navy technical-professional environments, and the program thus forms a valid testbed upon which to examine the effects of inserting specific gaming elements into a training curriculum and, ultimately, to test the major premises of the tool. A second enabling objective was to develop a taxonomy of high-payoff game elements (e.g., situational realism, challenge, ambient stimulation) that can impact the instructional outcomes of these specific training objectives (Task 2). The two taxonomies – cognitive functions and game elements – were then linked in a crosswalk (Task 3) to derive optimal combinations of gaming element variations that would form specifications for a high-payoff serious game that could satisfy the designated training objectives and thus be integrated into the competency areas selected in Task 1. The crosswalk will also form the organizing structure for the game-based training tool, TARGET. In this capacity, it will initially draw from existing empirical research in the serious game literature, as well as relevant theories and research in instructional multimedia and general instructional design. The tool will later be refined as we integrate findings from the empirical research we plan to conduct in Phase II.

To establish a proof-of-concept of the tool's validity, we exercised the crosswalk to generate an optimal set of design specifications and preliminary storyboards to achieve the training objectives of one of our competency areas, FMS operation (Task 4). These specifications are based both on pedagogical principles, content-validated with instructional design experts at ASU-AMT, and on current game design conventions, confirmed through discussions with an experienced game designer who will be on our project staff in Phase II. These specifications were then instantiated in a series of PowerPoint storyboards that contained a functioning FMS prototype, example instructional material, layered feedback, iconic gaming frames, a "virtual" captain to promote realism, an embedded series of arcade "gamelets" designed for high-speed, repetitive practice on key FMS part-tasks, and a host of other features to instill a true game-like atmosphere. A formative evaluation of the storyboards, conducted with experienced Mesa Air pilots (an intended industry-target user of our FMS and callout games), yielded extremely positive reactions to the concepts, with more than 95% of possible responses scored as "extremely favorable" or "favorable."

Because our ultimate goal is to transition our GBT technology from ASU-AMT to Navy technical environments as quickly as possible, a high-powered methodology is needed that will guide this application. ASI has a proven method, using the framework of reusable Design Patterns (Dick & Spiker, 2006), to serve this purpose. Developed originally in architecture and software, ASI is now using its own trademarked version called *Design Pattern Extraction and Migration.* The FMS programming game specifications and storyboards were analyzed in detail, and some 50-odd design patterns were identified/extracted. This analysis is being presented at the June 07 Software Engineering and Research Practice (SERP) conference (Mautone, Spiker, & Dick, 2007). As part of this analysis, we specify how the various 'design patterns' which make up the FMS programming game can be immediately applied to support navigator training in the Navy's landing craft air cushion (LCAC), an amphibious assault vehicle. We already have a claimant from the LCAC

community ready to receive and use this application once it is developed in Phase II; details are provided in the transition plan, Section 2 on DoD customers. The marriage of our GBT with the design pattern methodology give us, we believe, a very unique and powerful framework for rapidly developing and infusing high-payoff game elements into receptive Navy training environments.

In Task 4, we also outlined an evaluation plan describing how we will conduct a series of tightly-controlled studies of whether and how infusion of GBT into the ASU-AMT training curriculum impacts training effectiveness for FMS operation and procedure-based callouts. The close working relationship with ASU-AMT will allow us to directly control their training regime for the purposes of this study so that non-game control group subjects will receive the exact same training as the treatment game, minus the “game” aspects of our (1) FMS and (2) procedure-based callout prototypes. This setup will yield one of the cleanest and most clear-cut studies ever conducted on the impact of GBT (serious games) on training effectiveness. As a second research strand, we will conduct a series of smaller, controlled studies looking at the impact of individual or small combinations of game element variables on training performance. These will be conducted using a combination of undergraduates at ASI facilities and actual Mesa Air line pilots at ASU-AMT.

If the research we propose during the Phase I-II interim is funded, we will develop portions of the FMS programming game and conduct proof-of-concept empirical tests of its training impact. Transition funding will be used to develop several levels of the game and then conduct trial-runs with representative users. We will also use this period to introduce our GBT prototype to several of the Navy claimants identified in Section 2 of the transition plan. These will include the LCAC navigator community mentioned above along with key representatives from the submarine, surface ship, and UAV communities. In each case, we will use this transitional period to “get the word out” concerning our approach, solicit support, and begin the process of collecting specifications and requirements so that we may extract the appropriate design patterns to ensure timely development of GBT using a combination of Phase II, Phase II Enhancement, and Phase III funding.

We now describe the main accomplishments made during the Phase I period in more detail, beginning, in Chapter 2, with a description of the development of the game element taxonomy. In Chapter 3, we outline the methodology and findings of the cognitive task analysis we conducted of the ASU pilot training environment, and describe two training competencies that we selected as good candidates for game-based training. In Chapter 4 we describe how we combined the game taxonomy and cognitive task analysis findings in order to form the “Game Element – Training Competency Crosswalk” and how the crosswalk was applied to select game element variations best suited for a game targeting one of the selected competencies, FMS programming. In Chapter 5, we describe the development, specifications, and evaluation of the FMS game in more detail, and in Chapter 6, we describe the specifications of the game designed to target the second selected training competency, Profile-based Call outs. Chapter 7 outlines how the games and the principles behind the game design, could be applied to other training domains, and Chapter 8 reviews our plans for conducting empirical research in Phase II. In Chapter 9 we discuss additional feasibility issues, and in Chapter 10 we summarize our conclusions and recommendations. Finally, in Chapter 11, we provide a summary of the proposed Phase II work.

Chapter 2: Development of a Taxonomy of Serious Game Elements

We began our research by developing a taxonomy of serious game elements, based on a review of game-based instruction literature, as well as research on general instructional design principles, multimedia instruction, and cognitive psychology. The taxonomy is an attempt to identify, abstract, define, and categorize some of the more salient serious game elements and their multiple variations or “levels”. Profiling serious game elements in the form of a taxonomy serves several purposes. First, it allows us to describe games in a more systematic, well-defined manner. One of the major problems with the existing research literature on the effectiveness of game-based instruction is that the researchers often failed to provide a clear description of the specific characteristics of the games used in their study (Hays, 2005). Without a clear description of the specific elements that were included in the study games, it is difficult to draw any useful conclusions about which factors contributed to the success or failure of the game to improve learning outcomes. Secondly, the taxonomy provides a systematic means by which to select which game elements would be most appropriate to include in the design of a game created to target specific training objectives. The taxonomy, in essence, is a critical component in the development of a crosswalk linking training

environments, and specific gaming elements. The crosswalk, in turn, forms the foundation for the development of the game-based instruction decision-making tool, TARGET. Third, the taxonomy outlines a set of variables we may want to manipulate in the design of our Phase II empirical studies in which we test the premises of the tool.

Literature review

In order to create the taxonomy, we gathered and reviewed several articles related to game-based instruction. When we began our search, we were primarily interested in empirically-based articles that, 1) defined and manipulated game elements; 2) measured performance outcomes for retention, transfer, motivation, and factors other than improvement on the game itself; and 3) were relevant to adults. We also tried to focus on articles that referenced skills relevant to Navy training and the cognitive functions that we were outlining in our target training environment, identified in our task analysis of our proposed test-bed (which is described in more detail in Chapter 3.) Articles that referenced improved efficiency of training (e.g., reduced training time or resources) were also of interest. The majority of articles we examined were from peer-reviewed journals, review articles, and book chapters.

Our first objective in reviewing the literature was to distill a list of key game elements and possible variations in order to help us construct the serious game element taxonomy. Ideally, we also wanted to use the literature findings to help us begin constructing a crosswalk linking game elements and cognitive functions; thus, as noted above, we primarily looked for articles that described how manipulations of specific game elements affected cognitive and motivational outcomes.

We began with comprehensive review articles, (e.g., Fletcher & Tobias, 2006; Hays, 2005; Leemkuil, de Jong & Ootes, 2000; O'Neil, Winess, & Baker. 2005;), some of which were obtained through a search of article databases such as PSYINFO. Others were referred to us by our TPOC and by our instructional design SME, Dr. Richard Mayer, Professor of Cognitive Psychology at the University of California, Santa Barbara. The review articles frequently referred to definitions and characteristics of games, and thus provided a starting place for creating the main structure of the taxonomy. More importantly, however, the review articles were used to increase the efficiency of our search by leading us to articles that appeared to be empirically based and which described and tested game variations and their impact on learning outcomes. These types of articles were few and far between, however, so we expanded our search to also include studies which examined learning outcomes for game vs. no game groups. As we progressed in the search, we also loosened the criteria to include research studies which examined modifications to specific game elements, but which used game performance itself to measure performance improvements; although these types of studies were not what we were initially looking for, they did offer insight into how game elements can be varied to produce performance changes.

Though many of these articles often did not describe the game in enough detail to extract which specific game element variations affected training outcome (e.g., they referenced the use of feedback, but did not describe what type of feedback or how it was administered), others did provide such detail and thus provided information to help us fill in the taxonomy.

Construction of the Serious Gaming Elements Taxonomy

In conducting the literature review, we thought that rather than just summarize individual studies, it would be more efficient to create the structure of the taxonomy and fill it in with findings from the literature. We began by constructing a preliminary taxonomy after reviewing several of the review articles and distilling some of the major game element dimensions (such as challenge, goals, and feedback). We initially entered the higher level game element categories or dimensions into an Excel spreadsheet, and then broke each one down into subcategories, listing the possible ways in which the game element dimension could vary. For example, FEEDBACK could be presented in a number of ways (quantitative, qualitative), at varying intervals (immediately, periodically, after a sequence of events, etc) and at a variety of depths (e.g., detailed prescriptive feedback vs. brief confirmatory, correct/incorrect feedback). This preliminary taxonomy then served as a scaffold upon which to add findings from our literature search. As we added new game element variations and categorized them according to our preliminary matrix, we began to refine the taxonomy, adding categories and combining others. For example, we decided that, as we were focusing on serious game elements, game variations related to instructional support needed to have their own higher-level

category. In essence, we looked for looked for patterns that would allow us to categorize gaming element variations in an organized manner that would later allow us to describe and select specific gaming elements for inclusion into our test-bed training programs. In designing the taxonomy, we also conferred with our expert consultant in the field of instructional design, Dr. Richard Mayer. It should be noted, however, that the serious game element taxonomy, is still a work in progress and will be modified and refined as we gather new information – both from the planned Phase II research studies and from future research and feedback from other experts in the serious game field.

Taxonomy of Serious Game Elements

Figure 1 shows a schematic summary of the preliminary taxonomy of serious game elements, which displays the eight major categories or dimensions (Feedback, Goals, Rules and Constraints, Situational Realism, User Control, Challenge, Social Interaction, and Instructional Support and Structure), as well as subtypes and examples for each category. The subtypes reflect some of the boundary, temporal, and other dimensions upon which each serious game element category can vary. This includes *form*, which generally refers to different formats, types, or manifestations of the element, such as quantitative vs. qualitative feedback or cognitive vs. physical realism; *level,* which generally refers to the degree of detail, explicitness, intensity, or complexity of the element, such as prescriptive vs. confirmatory feedback, or well-defined vs. ill-defined goals; *temporal factors*, which generally refer to aspects such as frequency and immediacy, as well as how dynamic the variable is; and *procedural factors,* which refer to how the elements are presented.

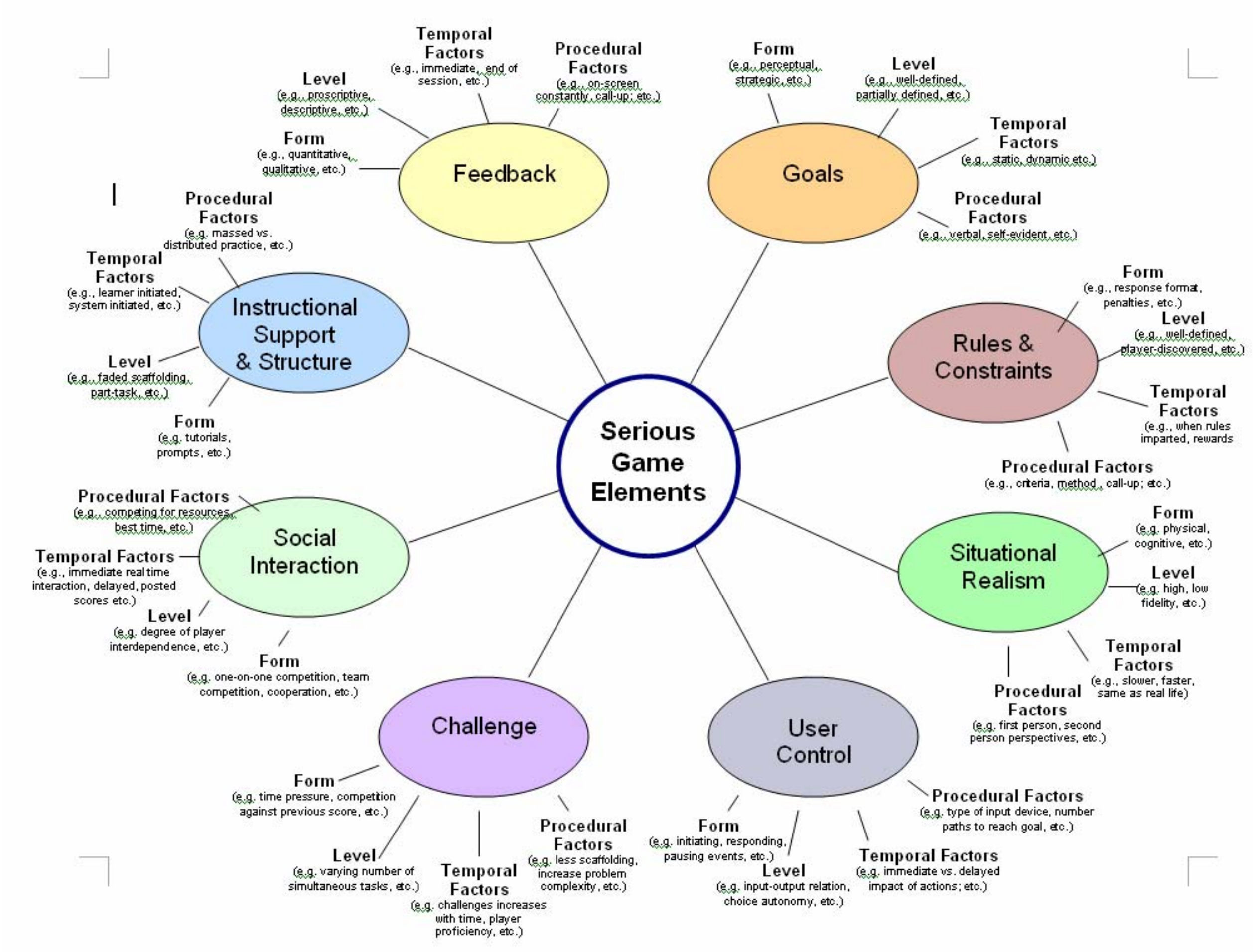


Figure 1 - Schematic Representation of the Serious Game Element Taxonomy

Below is a more detailed description of the components of the taxonomy. For each of the major game element categories, we provide:

- a general **overview** of each of the game element category
- associated ***instructional issues*** – how the game element impacts learning
- associated ***motivational issues*** – how the element relates to the game's sense of "fun"
- a list of ***sub-types and sample variations*** that fall under that game element category, and
- a brief example of a ***research study*** illustrating the effects of that particular game element.

1) Feedback

Feedback refers to information provided to the player about progress toward the game's goals. Feedback has two primary functions: to verify and to elaborate (Cameron & Dwyer, 2005). It can vary from a simple confirmation that a particular action or choice was correct, to a simple percentage correct score, to elaborate, proscriptive information about what the player did correctly and incorrectly, how far they are from optimal performance, and what changes they need to make in order to improve performance and reach the goal.

Different training situations may require different feedback variations. For example, in situations where rapid responses are required, providing quantitative, event-specific feedback is likely to be more appropriate than feedback that requires more processing by the player and, thus, may disrupt the player's flow of responses, causing frustrations and affecting both learning outcomes and player motivation. There are also cost factors to consider. It is likely to be more labor intensive for instructors to prepare prescriptive qualitative feedback. If research shows that providing less elaborate feedback for a particular target task is sufficient, that knowledge could reduce the amount of time and money needed to develop the game.

Feedback Sub-Types

Format Factors

- ❑ quantitative (e.g. a number of items correct, degrees off target, time to complete task)
- ❑ qualitative (e.g. verbal commentary or text paragraph describing what was correct or incorrect)
- ❑ natural consequence (e.g., seeing the car crash or plant die)
- ❑ graphical (e.g., seeing the ideal trajectory compared to yours)
- ❑ combination of two or more of the above

Level/Intensity Factors

- ❑ prescriptive (detailing why action/response was wrong and how to improve)
- ❑ confirmatory/descriptive (whether action/response was correct or incorrect)
- ❑ elaborative (detailing why the correct response is the best choice)
- ❑ reflective (requiring players to explain why they chose their response)
- ❑ general level (pass/fail)
- ❑ combination of two or more of the above

Temporal Factors

- ❑ immediate (e.g., 2-5 seconds after each action/response)
- ❑ periodic – player independent (e.g., every 5 minutes)
- ❑ periodic – player dependent (e.g., after player completes sequence of actions or reaches subgoal)
- ❑ end of game
- ❑ combination of two or more of the above

Procedural or "How" Factors

- ❑ constantly displayed
- ❑ periodically displayed
- ❑ displayed upon player request

- ❑ cumulative or event-specific
- ❑ combination of two or more of the above

Examples of a study in which feedback was examined or manipulated: Cameron & Dwyer (2005) conducted a study in which type of feedback given during the course of an instructional game was manipulated. The instructional goal of the game used in the study was to provide students with an opportunity to rehearse what factual and conceptual knowledge that they had previously learned about how the heart works. As students interacted with the game, they were asked questions. Some received no feedback about their performance on the questions, others were just told whether their response was correct or incorrect, and others were given elaborative feedback, explaining the correct answers. As expected, those who received elaborative feedback performed better on delayed retention tests. While it would appear that elaborative feedback is superior to other forms of feedback, particularly in the training of conceptual knowledge, in other training situations, other, less intensive feedback may be sufficient.

2) Goals

The category of goals refers primarily to the underlying targeted training goals rather than just the apparent goal of the game – which may or may not be identical to the training goal. For example, in a game that involves solving puzzles and navigating a maze in order to reach a treasure, the apparent goal is to reach the treasure, whereas the training goals are to improve problem-solving skills, spatial awareness, and navigation skills. As can be seen below, some of the training goal features relate closely to Bloom's Taxonomy of Educational Objectives.

It should be noted that the characteristics of the goals employed in a game not only impact learning, they also affect players' level of engagement. For example, goals that are too complex – or too simple – are likely to reduce the player's enjoyment of the game. Likewise, games that employ goals that are motivating and relevant for the player (i.e., the player can relate to the goal and see its purpose or utility) are also more likely to be considered fun. Providing clear goals, presenting an overriding goal early, and incorporating short-term goals throughout play are also key factors affecting game playability (Desurvivre, Caplan, and Toth, 2004).

Goal Sub-Types

Format Factors

- ❑ factual (e.g., learning terms)
- ❑ conceptual (e.g., learning how parts of system are related; diagnosing problems)
- ❑ procedural (e.g., learning how to operate a piece of equipment,
- ❑ strategy (e.g., developing skills in managing attention or in handling high workload demands)
- ❑ perceptual motor (e.g. speed and accuracy)
- ❑ problem-solving
- ❑ spatial/navigation
- ❑ combination of two or more of the above

Level/Intensity Factors

- ❑ well-defined (clear to the player when game begins)
- ❑ partially defined (initial goals clearly defined when game begins; subsequent goals, less so)
- ❑ ill-defined (goals only vaguely defined; player needs to discover them as they go along)
- ❑ player defined (player chooses which goals to focus on, e.g., improving speed vs. accuracy)

Temporal Factors

- ❑ goals are static (stay more-or-less the same as the game progresses)
- ❑ goals are dynamic, player-dependent (goals change as player's proficiency progresses)
- ❑ goals are dynamic, player-independent (goals change based on time or other factors)
- ❑ combination of two or more of the above

Procedural or "How" Factors

- ❑ goals are imparted through written or verbal instructions
- ❑ goals are self-evident (e.g., shooting at targets)
- ❑ specific – providing details on exactly what the player needs to do
- ❑ standard – providing a general

Examples of a study in which the game element, Goals, was manipulated: One study in which the effect of manipulating goal specificity on learning outcomes was examined was conducted by Miller, et al. (1999). In this study, college students learned physics principles related to electrical charge interactions while playing one of three versions of an electronic field hockey game. The game involved manipulating electrically charged objects along certain trajectories via interactions with other electrically charged objects. In the no-goal version, students were allowed to play with the simulation but were given no specific tasks; in the standard-goal condition, students were told to try to get one of the objects through a "net"; and in the specific-goal condition, students were given a well-defined goal of where exactly to place the obstacle and were shown the ideal trajectory. They were also given feedback about how close they were to the goal. Those in the specific-goal group and no-goal conditions did better than those in the standard-goal on measures of qualitative physics knowledge; interestingly, the no-goal group also did well on the assessments. This study highlights the effect that the level or specificity of goal definition can have an impact on learning outcomes.

3) Rules and Constraints

Rules and Constraints refers to how the boundaries of the game are imparted to the player, and what the consequences are for adhering to or violating these boundaries. They encompass such factors as the instructions for play, range of allowable actions, rewards and penalties, and general path of play. As with goals, the characteristics of the rules and constraints used in a game affect both learning outcomes and players' enjoyment of the game. If, for example, players are repeatedly penalized for failure on the same task, without any opportunity to pursue a different path, practice the requisite skill, or obtain remedial help, then the player is likely to stop playing the game.

Rules and Constraints Sub-Types

Format Factors

- ❑ response format: motor, keyboard, verbal, etc.
- ❑ type of rewards and/or penalties for adhering to rules (resources, time, points)
- ❑ consequences of success/failure (e.g., elimination from game; promotion to next level)
- ❑ rules about possible actions
- ❑ rules about when to carry out possible actions and conditions that must be met
- ❑ rules underlying game play (e.g., if you do X, Y will happen)
- ❑ rules about time constraints

Level/Intensity Factors

- ❑ player rules well-defined in advance (e.g., list of allowable actions, subgoals, etc.)
- ❑ some player rules discovered as play progresses (e.g., player tests the rule boundaries)
- ❑ transparent rules governing underlying game play (e.g., cause-effect mechanism clear)
- ❑ rules governing underlying game play not immediately apparent to players
- ❑ amount of rule cueing (i.e., degree to which rules need to be memorized beforehand)
- ❑ number of response options (e.g., number of tools, procedures, paths player may use)
- ❑ level of complexity (e.g., only one way to carry out a particular action vs. many)

Temporal Factors

- ❑ when and how rules are learned
- ❑ static or dynamic (e.g., changing response pattern or weapon depending on type of enemy)
- ❑ schedule for receiving rewards and penalties (e.g., fixed or variable; ratio or

Procedural or "How" Factors

- ❑ criteria for rewards and penalties
- ❑ method of disseminating rules to players
- ❑ degree to which rules adhere to reality

Examples of a studies in which the game element, Rules and Constraints, played a role: In a study investigating different decision-making tactics, (Lewis & Barlow, 2005), teams used a first-person shooter game to play out scenarios requiring dynamic decision-making (i.e., making rapid, sequential decisions, the outcome of which affects other decisions). Teams who scored higher were rated as using prior experience more effectively in the decision-making process, and having better Situational Awareness. This study was more about using games to investigate decision-making, but the authors did note that the rules of the game, in particular, the penalty structure, may have had an unintended effect on performance – specifically, low penalties for failure may have led to more risky/foolhardy decision-making. In another study, a game comparing the effects of using a drill vs. game format to train students to recognize and repair electrical circuits, Whitehill and McDonald (1993) manipulated rules regarding the reward payoff schedule for both groups – using either a fixed or variable payoff for correctly solving different types of problems. They found that, although overall performance measures for the game and drill groups did not vary, those in the game condition who received variable payoffs tended to persist longer at the task. This finding is in line with cognitive psychology research on the effects of different reinforcement schedules on the stability of operantly-conditioned response patterns.

4) Situational Realism

Situational realism refers to how detailed the storyline, context, environment, and general appearance of the game are, and the degree to which these factors correspond to real life. All of these factors can influence the level of engagement a player experiences while playing the game and affect instructional outcomes. The main premise is that the greater the overlap between specific characteristics of the game (e.g., setting, player roles and activities, rules, etc.) and specific instructional objectives, the more instructionally effective the game will be (Hays, 2005). Likewise, the more immersed in the storyline and the more players identify with their role in the game – something that those in the game development industry refer to as "character investment" – the more likely they are to find the game fun and motivating.

Situational Realism Sub-Types

Format Factors

- ❑ physical realism (the game environment looks and sounds like the actual training environment)
- ❑ cognitive realism (the game places similar cognitive demands, such as attention allocation, working memory load, strategy development, etc., as the actual training environment)
- ❑ behavioral realism (the player and avatars interact as they would in real life)
- ❑ emotional realism (emotions and stress evoked are similar to that evoked in actual environment)
- ❑ storyline realism (the degree to which the game's purpose, goals, and roles are similar to real-life)
- ❑ sensory detail (e.g. the level of visual and auditory detail – whether fantasy or reality)
- ❑ the degree to which the invisible is made visible (underlying processes more transparent)
- ❑ a combination of two or more of these factors

Level/Intensity Factors

- ❑ degree of fidelity (high, medium, low)

Temporal Factors

- ❑ timing fidelity (do events occur at a slower or faster pace than real life
- ❑ degree of flexibility (does fidelity vary dependent on player's proficiency)

Procedural or "How" Factors

- ❑ whether the game presents first person or second person perspectives

- ❑ how well the actual tasks correspond to the real life training environment
- ❑ how well allowable actions and input/output devices correspond to real life
- ❑ whether the environment responds similarly to the way it would in real life

Examples of a study in which the game element, Situational Realism, affected learning outcomes: Traditionally, the focus in creating situational realism was on physical similarity. It was believed that the more a game's physical appearance corresponded to the real life training scenario, the better potential it had for improving performance on real-world transfer tasks; however, recent studies in both gaming and multimedia show this is not always the case and that cognitive and behavioral realism are likely to play an even more important role. In a study by Gopher, et al. (1994), future pilots played the game, Space Fortress, which was designed to simulate the cognitive load and attention management demands that are experienced during flight. The physical characteristics of the game, however, were very simple; the screen and controls did not resemble the interface and controls of an airplane. Nonetheless, those who played the game later performed much better on measures of flight performance. Furthermore, in another study in which pilot trainees were given either Space Fortress or an off-the-shelf flight game with higher physical fidelity, only those in the Space Fortress game-playing condition performed better than the control group (Hart and Batiste, 1992). According to the authors of these studies, it was the cognitive realism, or the degree to which the game placed similar attentional and cognitive load demands upon the players, that contributed to transfer. Along similar lines, multimedia studies have shown that, for beginner learners at least, simple line drawings are more effective than high fidelity diagrams, photographs, or animations in teaching students how a system works (Mayer, 2005).

5) User Control

According to Hays (2005), when used in reference to gaming, control does not refer to a player "being in control" but rather to the player being able to exercise control and influence the outcome of events. Player control is one of the factors most often cited as contributing to motivation and enjoyment of the game. In this taxonomy, it refers to the level of autonomy that a player has in terms of which actions to carry out and when, and the degree to which events are contingent upon a player's actions. It includes such things as number and variety of actions a player can choose from, as well as factors such as whether players are able to see the effects of their actions, or are permitted to choose multiple paths to reach a particular goal. Games that allow players to "learn by doing" – as long as the tasks are relevant to the intended training objectives – are more likely to result in better learning outcomes (Aldrich, 2005). They are also more likely to engage the player, provided the actions and choices are at the appropriate level of difficulty (i.e., match the player's skill level.)

User Control Sub-Types

Format Factors

- ❑ initiating actions and events
- ❑ responding to events
- ❑ controlling outcome of an event
- ❑ setting parameters and then watching events unfold
- ❑ controlling the pacing
- ❑ controlling repeats and pausing
- ❑ manipulating tools and devices

Level/Intensity Factors

- ❑ how much the player can control at any given time
- ❑ whether there is requirement to switch response choices for a specific event
- ❑ whether the player can select which action to take (e.g., use of weapons)
- ❑ input/output relationship (1st order, 2nd order, etc.)
- ❑ continuous vs. discrete (player is constantly directing action or watching and responding to it)

Temporal Factors

- ❑ whether the number of choices available increases with time or player proficiency
- ❑ immediacy of seeing the results of one's actions
- ❑ proactive vs. reactive

Procedural or "How" Factors

- ❑ number of response options
- ❑ number of paths or methods to reach the goal
- ❑ number of choices at any given time
- ❑ type of input device (joystick, keyboard, etc.)

Examples of studies in which the game element, Control, was a factor: White and Fredericksen (1998), conducted a series of studies using ThinkerTools, a simulation game designed to teach physics principles using a computer interface that provides users with a great deal of control. During the game, users perform physics experiments in a variety of conditions, observe outcomes, and compare how the outcomes fit with their initial hypotheses. In one study, middle school students who interacted with the game performed better than high school physics students on subsequent test questions involving the application of basic Newtonian principles.

6) Challenge

Challenge refers to the obstacles that stand in the way of reaching a goal – whether the goal is one addressed directly in the game (e.g., mastering a skill and reaching another level), or a personal goal (e.g., being the highest-scoring player) – and is a critical factor affecting player motivation. Introducing challenges into a serious game are particularly important when some of the skills targeted by the game's training objectives may be inherently boring. In these situations, in order to engage players and encourage them to practice enough to master these skills, the game must build in motivating challenges by introducing such elements as competition, uncertainly, and multiple levels of graduated complexity. Competition, whether It be with other players, the clock, a virtual player, or even the player's previous performance is a powerful motivating factor (Aldrich, 2005). Introducing a degree of uncertainty – about whether they will be able to successfully accomplish the tasks under a particular constraint (e.g., complete all steps within a given time frame), as well as about what might happen next (e.g., introducing twists) – also contributes to how challenging players will find the game. In order for a game to be challenging for a wide range of players, with varying levels of ability, it should incorporate multiple levels of increasing difficulty and complexity. This format allows players to progress at their own pace, so adept players are not bored and slower players are not frustrated.

Challenge Sub-Types

Format Factors

- ❑ competition against oneself (e.g., previous scores, fastest times, etc.)
- ❑ competition against other players
- ❑ competition against time (completing something within a specific time period)
- ❑ competition against criterion scores
- ❑ time pressure
- ❑ number of tasks
- ❑ number of simultaneous demands
- ❑ integration of tasks
- ❑ unexpected events

Level/Intensity Factors

- ❑ low challenge (e.g., fewer options, fewer obstacles, slower pacing, less complex problems)
- ❑ moderate challenge
- ❑ high levels of challenge

Temporal Factors – what initiates increases in challenge

- ❑ player's proficiency (e.g., speed and accuracy on a particular task)
- ❑ completion of previous tasks/levels
- ❑ meeting specific criteria

Procedural or "How" Factors – what changes when player moves to the next level

- ❑ increased pacing
- ❑ increased number of choices
- ❑ more complex information or tasks
- ❑ less scaffolding
- ❑ more tasks to perform simultaneously

Example of a study in which the game element, Challenge, was a factor: Garris, et al. (2001) conducted a study in which students played the game *Bottom Gun*, which was designed to improve visual skills, by providing practice with activities such as making estimates of critical visual variables, Both the game and the control group performed the same tasks and received the same type of feedback, but the version used in the game condition incorporated many of the characteristics that add to the element of challenge – characteristics such as elements of danger and uncertainty, competitiveness, and increasing difficulty. Those who played the game version showed improved performance and significantly fewer errors than those in the control group.

7) Social Interaction

This category refers to how – or whether – players interact with, communicate, and cooperate (or compete against) each other, and with virtual characters in the game. As with challenge and situational realism, this game element can also greatly affect how immersed and invested the player becomes in the game. From a cognitive perspective, introducing social processes also has an impact on learning. Research in multimedia learning has shown that personalizing instructions (e.g., saying "your lungs" instead of "the lungs") results in improved retention and transfer (Mayer, 2005).

Social Interaction Sub-Types

Format Factors

- ❑ number of other players (from 0 to thousands)
- ❑ 1-1 competition
- ❑ 2-1 competition
- ❑ team competition
- ❑ 1-1 cooperation
- ❑ team cooperation

Level/Intensity Factors

- ❑ how interdependent players' actions are (if one's actions affects another's)

Temporal Factors

- ❑ communication among players occurs in real time (vs. near real time vs. after a longer delay)
- ❑ communication occurs periodically
- ❑ posted scores
- ❑ immediacy of communication (verbal, textual, etc.)

Procedural or "How" Factors

- ❑ comparing final outcome scores
- ❑ competing for best time
- ❑ competing for zero-sum resources

Examples of a study in which the game element, Social interaction (in the form of competition), was manipulated: Fisher (1976) manipulated different types of challenge, in the form of competition formats, in a game designed to teach vocabulary skills. College students played the game, *Dictionary*, either against one other competitor (1-1), two other competitors (1-2), or in a three-person team competing against another three-person team. All three game groups performed better on subsequent tests than did the control group, with the 1-2 group achieving the highest scores. Although a relatively simple study, the results indicate that different types of competition can have differential effects on learning.

8) Structure and Instructional Support

This category refers to the adjunct instructional support, such as hints, pre-training, and focus questions, that help learners understand how to use and learn from the game, as well as other key instructional factors such as how practice sessions are structured. While this category would, at first seem to mainly affect learning outcomes, it is also a critical factor in affecting player motivation. Current theories on motivation and cognition emphasize that one of the most effective ways to motivate learners and make a lesson fun isn't to dress it up with irrelevant "bells and whistles" (which can actually hinder learning), but to help students make sense of the material (Kintch, 1980; Wade, 1992; Mayer, 2003)

Structure and Support Sub-Types

Format Factors

- ❑ tutorials
- ❑ prompts, hints, tips, coaching – dependent on player performance (e.g., intelligent tutoring)
- ❑ prompts, hints, tips – independent on player performance (e.g., all player receive same hints)

Level/Intensity Factors

- ❑ progressively more complex
- ❑ part-task (focusing on mastering the subtasks before encountering the full game environment)
- ❑ emphasis change (encountering the whole task environment at all times, but with changing emphasis on particular tasks)
- ❑ modeling
- ❑ fading out scaffolding
- ❑ providing hints, coaching, tips, prompts
- ❑ specifically related to player's performance (intelligent tutoring)
- ❑ general (everyone gets the same hints)

Temporal Factors

- ❑ aid provided when the learner requests it (vs. system initiated)
- ❑ prior to a session
- ❑ when player makes mistake/has trouble
- ❑ pre-questions

Procedural or "How" Factors

Procedural factors related to Pre-training/tutorials

- ❑ going over potentially unfamiliar terms and concepts prior to the game
- ❑ directions on how to interact with the game
- ❑ outline how game relates to training objectives

Procedural factors related to Practice structure

- ❑ number of repetitions
- ❑ time between repetitions (inter-trial interval)
- ❑ criterion references or remedial
- ❑ massed vs. distributed
- ❑ independence of levels: whether higher levels require applying skills learned earlier

Procedural factors related to the format in which instruction is conveyed

- ❑ pictorial
- ❑ strategic
- ❑ verbal

Example of studies in which the game element, Instructional Support, was manipulated:

Although there were confounds in their study design Gopher, et. al (1994), found that providing students with tips and individual tutoring was not necessary for improving performance on the criterion test (flight performance). Interestingly, those who received emphasis-change instructions (without tutoring) didn't do as well on the game itself as those who received part-task training accompanied by hints and individual feedback; however, they did just as well on subsequent, real world flight performance measures. This finding illustrates the limitations of making assumptions about the effectiveness of game element manipulations based solely on improvements in game performance rather than on transfer tests.

In another study, Mayer, Mautone, and Prothero (2002) found that different types of scaffolding may be more appropriate for different training needs. In this study, college students played a game in which they gathered and interpreted spatial, line-drawing data (similar to sonar patterns) in order to identify hidden geological features. In a series of controlled studies, students were given either pictorial scaffolding, which provided aid in interpreting the visual data, or verbal scaffolding, which provided aid in developing strategies for gathering data. Results indicated that for this particular training objective – developing skills in interpreting spatial data – pictorial aids given prior to the task resulted in better performance not only on the game task itself, but on subsequent transfer tests.

These studies highlight the importance of conducting a cognitive task analysis of the training objectives, targeting the tasks students have the most difficulty with, and inserting game elements that are the best candidates for improving performance for that particular training objective. The game element taxonomy serves as one of the key cornerstones in the development of a Game Element – Training Competency crosswalk that matches training needs with appropriate game elements.

Chapter 3: Development of the Cognitive Functions Taxonomy

In addition to the game taxonomy, the other cornerstone of the Game Element – Training Competency crosswalk is the taxonomy of cognitive functions associated with common training objectives. We began development of the taxonomy by analyzing the training environment of ASU's flight training program in order to identify training needs. The goal was to identify core training competencies that:

- hold the most promise for GBT enhancement,
- are representative of a range of competency types,
- are similar to Navy tasks, have measurable outcomes,
- can be adapted to include GBT elements, and
- represent current training needs.

From this analysis, we identified two core training competencies as the best candidate for game-based training: FMS Operation, and Profile-Based Callouts. We then conducted a task analysis of the two selected competencies and developed a taxonomy of cognitive functions for each.

The ASU-AMT Training Environment

We chose to focus on our initial task analysis on the pilot training program in ASU's Aeronautical Management Technology (AMT) department. AMT offers a unique and highly controlled environment in which GBT concepts may be identified, developed, implemented, and validated. With a current enrollment of 250 students, AMT provides its undergraduates with an intensive, four-year program of focused study in aviation that prepares its graduates to fly with US regional and major airlines. Students who successfully

complete the program are guaranteed an interview for a First Officer position with Mesa Air Group, ASU's collaborative airline, and many of ASU's graduating seniors do go on to become pilots for Mesa Airlines.

As one of only 18 institutions accredited by the Council on Aviation Accreditation, ASU-AMT places nationally in the top tier of more than 95 four-year institutions with aviation programs. While the bulk of AMT's students are in the Professional Flight BS Program, others are enrolled in the Air Transportation Management, Applied Science, and Technology BS programs. Collectively, these programs constitute a substantial population of aviation-knowledgeable subjects for exploring new game concepts and conducting experiments. Because AMT does not presently utilize GBT elements in any of its curricula, its current training flow will serve as a clean baseline of "conventional training" against which selective introduction of GBT elements may be compared.

AMT has been successful in preparing its students for a career in aviation, where its administration is promoting a research-driven curriculum development to support its Airline Bridge Training Program. This program is designed to provide students with the requisite skills to be a successful airline pilot upon graduating. This program is oriented toward cognitive and behavioral skill development, coupled with instilling professional attitudes, to effect strong transfer to the airline environment. The department is continually striving to find new and improved ways to promote skill development and further enhance skill transfer, so their interest in exploring GBT is very high.

The training curriculum confronts the student with a progressive, challenging array of classes designed to instill and reinforce the necessary KSAs to safely fly a regional jet airliner. Starting with coursework in ground school, meteorology, power plants, and safety regulations, students move to system-specific courses that allow them to receive their private pilot's license, commercial license, instrument rating, multi-engine and multi-crew rating, and flight instructor certification. Accomplishing all this in four years is challenging, and the department is searching for ways to accelerate learning, encourage self-study, facilitate practice, and increase core skills.

On the technology side, the department houses an array of computers, simulators, and learning stations that can be programmed with GBT elements of virtually any type. Their learning lab has 20 CBT stations for operating the flight management system (FMS) and other individual aircraft systems (radios, GPS, weather radar). The instrument flight lab has eight computers that serve as PC-based aviation training devices (PCATDs). These mini-simulators allow students to link classroom lessons with hands-on skill practice where flight control devices (yoke, throttle) are used rather than a mouse or joy stick. AMT also has an advanced briefing room with two-dimensional screens where students may practice their call-out procedures and "flows" (i.e., the intricate and scripted information exchanges between pilots and co-pilots during discrete flight segments).

On the upper end of technology, the department houses two multi-million dollar Level D flight simulators, a cadre of cockpit familiarization trainers, a human factors research laboratory, part-task trainers, and physical mockups. In addition, one of the most exciting and useful settings is AMT's Level 5 Flight Training Device (FTD). The FTD is used for the students' final 40-hour (10 4-hour sessions) training that they receive during their last semester in the program. The FTD provides an ideal criterion environment in which the training of core KSAs may be assessed.

From past projects with AMT, Anacapa has become familiar with their curriculum, students, and instructors, which greatly facilitated our review and analysis of the pilot training program.

Curriculum Review

Our review of the ASU-Mesa training curriculum was conducted in two parts. We first examined the department's course offerings to identify places where key competencies are taught, specific instructional equipment used, and areas where technical demands on the students would be significant. During the first two years, students spend the majority of their time taking coursework to fulfill basic university requirements (e.g., English, Calculus, etc.) and are enrolled in classes that support the ambitious series of pilot licenses they are expected to obtain (i.e., private pilot's license; instrument rating; commercial license; certified flight instructor, certified flight instructor instrument, and multi-engine rating).

Courses supporting the in-aircraft flying that students will be engaged in include ground school, meteorology, aircraft powerplants, and flight safety, among others. Because there is almost no coursework devoted to flying commercial jet aircraft during the first two years of the student's curriculum, we directed our analysis to the last two years (Junior, Senior). With this focus, there are 12 classes that are of primary interest to us. These are:

Flight Safety IV (AMT 400)	Aviation Safety & Human Factors (AMT 410)
Aviation Law & Regulations (AMT 442)	*Airline Instrument Procedures (AMT 482)*
Aircraft Design & Logistics Management (AMT 350)	*Regional Jet Aircraft Systems (AMT 486)*
Air Navigation (AMT 396)	Airline Administration (AMT 489)
The Aviation Professional (AMT 396)	*RJ Operations Capstone (AMT 490)*
National Aviation Policy (AMT 408)	Technical Communications (TWC 400)

Looking at the above list, it is apparent that not all of these courses have direct relevance to the acquisition of skilled competencies needed to fly commercial jets. That is, some of the courses are really designed to provide ancillary, supporting information (knowledge) as opposed to the actual skills required to be a successful commercial airline pilot. With skill/competency as our focus, there are really four courses whose curricula are of greatest interest. These classes are denoted above in italics.

This review was followed by a visit to ASU-Mesa to meet with instructors, faculty, and associated line pilots. Our discussions were specifically focused on the four key courses (AMT 396, 482, 482, 490) and the core competencies that students are expected to master before graduation. This visit, when combined with our prior knowledge of the ASU-Mesa training operations, gave us a fairly detailed map of the training flow that students experience during their tenure in the department. From this discussion, we were able to identify a list of distinct competencies (with a number of associated sub-competencies) as being potentially relevant for insertion of game-based training (GBT) elements. . They were selected based on the following six criteria:

1. It is frequently performed, i.e., will be needed on virtually every flight
2. It is technically challenging
3. There is an expressed need for better performance
4. There are recognized limitations in current training
5. It is important to the gaining unit (in this case, the airline hiring the student) that students have proficiency in the competency
6. It has clear analogs in Navy training curricula

We then reviewed the core competency list with three expert pilots at ASU-Mesa: the associate chair of the department, a senior lecturer in the department, and a senior instructor and line pilot with Mesa Air Group (co-located with ASU-Mesa). Collectively, these three individuals have amassed thousands of hours of operational flying experience, hundreds of hours in classroom and simulator training, and in-depth knowledge of the ASU-Mesa training curriculum.

Based on their review, it was agreed that two competencies, (1) flight management system (FMS) operation and (2) callouts and flow, would be the ones best-suited for our proof of concept development. FMS operation was unanimously agreed upon by our SMEs because students experience a number of problems using the FMS computer successfully during high fidelity simulator training later in their training. Typical problems include incorrect data entry, lack of mode awareness, and difficulties in visualizing the trajectory the aircraft will follow based on programmed input. Similarly, callouts and flow are a problem competency as students often fail to follow the scripted communications that are required between crew members during

selected profiles or segments of flight. Since mastery of both competencies is expected by the airlines who will hire these students, finding ways to improve their performance was deemed important by all of our SMEs. In the following subsection, we describe each competency in more detail, followed by a decomposition of the competency into key tasks, component subtasks, and cognitive functions.

Cognitive Task Analysis

With the two competencies thus identified, we next performed a mid-level cognitive ask analysis (CTA) on each competency. CTA is a collection of techniques, based on extensions of traditional task analysis, designed to yield information about the knowledge, thought processes, and goal structures that underlie observable task performance (Chipman, Schraagen, & Shalin, 2000). Specifically, we used a variant of the Consistent Component Method (Fisk & Eggemeier, 1988) to successively and iteratively decompose each competency into key tasks, component subtasks, and associated cognitive functions. For both competencies, the CTA was purposely slanted to identify behavioral manifestations of cognitive processes as that will help us pinpoint where, in the flow of each task, observers can be placed with measurement instruments during subsequent evaluations of task performance. In our initial analysis, our focus was on extracting component behavioral-cognitive activities that are (1) error-prone and (2) potentially modifiable via training. Below is a summary of our analysis of each of the two competencies.

Competency 1 – FMS Operation.

Overview:

The airplane's Flight Management System (FMS) is the pilot's primary interface to the software that controls the plane's navigation and performance inputs. All modern commercial jet aircraft rely on the FMS for accurate navigation, aircraft performance, and flight control. The CRJ 200 (the target aircraft of the ASU-AMT program) is no exception. Unquestionably, the FMS is the most important, and most complex, computer system the pilot must operate on each and every flight (Marrenbach & Kraiss, 2000). The system is responsible for flight planning, control of navigation sensors, set up of multi-function displays, radio tuning, fuel-efficient flight, and many other safety-critical functions (Rockwell-Collins, 1999). When properly configured and loaded, the FMS should significantly reduce pilot workload, allowing the flight computer to fly the plane on auto-pilot for virtually every minute they are airborne.

As can be seen in from a snapshot of the FMS shown in Figure 2, the FMS is a complex electronic device with a fairly non-intuitive interface that is quite different from the GUI interface most people are familiar with. As a system, the FMS consists of the flight management computer, control display unit (CDU), and a database unit.

Figure 2. FMS display.

Although the FMS is basically a computer, it does not operate with the simplicity of a PC nor does it have the iconic-oriented "drag and drop" functionality that most computer users have come to rely on. Rather, the FMS is a ruggedized, flight-worthy computer whose basic design was completed more than 20 years ago and which has been employed, virtually unaltered, for many years. Because modifications to software and hardware are prohibitively expensive (given the large numbers of aircraft involved), it is unlikely that we will soon see any improvements in FMS interface design or functionality, despite encouraging tests from inspired new conceptual designs (e.g., Marrenbach & Kraiss, 2000). In addition, since all military aircraft have their own, very similar, version of an FMS (including functionality for weapons deployment and other tactics), any GBT enhancements to training that facilitate FMS operation in the ASU-Mesa context will undoubtedly transfer to the military environment.

Before discussing the results of our CTA, let us first describe the basic aspects of FMS operation to illustrate its peculiarities and difficulty of use. Referring to Figure 2, the CDU contains two rows of six line select buttons on the upper left and upper right part of the display, a set of alphanumeric buttons at the bottom of the display, and a set of single-button activation mode keys in the middle. Represented by the brackets at

the bottom of the screen is the entry of what is called the “scratchpad,” i.e., the display results from the pilot entering data via the alphanumeric keys. Cursor control occurs through the up and down arrow keys below the display. Since many of the mode functions have multiple pages in their sub-menu structure, a pre-page and next-page keys will step the user through each page in the menu structure. When new information is to be entered, or some function (e.g., tuning a radio station) is to be actuated, the user must depress the Exec (execute) button so that action will initiate the underlying programming logic.

This command-driven interface is quite different from the GUI framework of personal computers, and not surprisingly, young student pilots find the required sequence of key operations (e.g., use of line-select keys, a separate execute key, very “deep” menu layers) quite unnatural. Some of the problems that are typically observed while students are learning to use the FMS include failing to hit the Exec key (leaving the system in idle), entering incorrect data, hitting the wrong key, failing to enter necessary information, not recognizing the particular “mode” the display is in, failing to detect when the system is processing erroneous data, and generally not being very adept or smooth at entering the required data quickly enough while under time pressure to get the system loaded before takeoff.

Cognitive Task Analysis:

While there are literally dozens of FMS functions and features described in the Collins FMS pilot’s guide, interviews with SMEs revealed that ASU-Mesa student pilots must master four basic FMS tasks by the end of their studies. These include

(1) Programming learning to program the FMS preflight, which includes verifying default information, entering the flight plan, setting up display functionality, and entering critical parameters pertaining to the plane’s weight, cargo, and fuel load as well as computed descent angles, wind speeds, temperature, and other critical variables that affect the navigation and performance of the aircraft

(2) Locating and updating information, such as accessing an existing flight plan to change the sequence of waypoints to be flown (e.g., in response to a request by ATC)

(3) Extracting the necessary to-be-entered data from the dispatch release form and other supporting material, and

(4) Recognizing and correcting errors and inconsistencies when there is a discrepancy in the FMS internal database.

For each of these four main tasks, a CTA was performed that identified anywhere from 6-11 component subtasks. Tables 1-4 provide the results of this decomposition. The left column of each table lists the component subtask, and the middle column provides a brief description of how that subtask is performed in the context of FMS operation. The right column lists the cognitive function(s) that underlie that component subtask. Please note that the results of this analysis, for both this competency and the callout-flow competency, have yet to be fully content-validated by our ASU-Mesa SMEs. Hence, it should be considered preliminary at this point. Content validation of the CTA results will be accomplished before the next report is issued.

***Task 1: Programming the FMS*.** The most labor-intensive FMS task involves loading (or “programming the device” as the pilots call it) the system with the data obtained from a number of sources, including the dispatch release form, communications with Air Traffic Control (ATC) and automated recording of current conditions (ATIS). The pilots must load the system quickly, usually within 2-5 minutes, while the plane is on the ground before beginning their taxi. Data entry has to conform to the mode organization of the FMS, where ASU-Mesa (and Mesa Airlines) encourages students-pilots to follow a specific sequence in order to verify and enter the required data. The pilot will first press the INDEX button, then a menu line select key in order to navigate to the STATUS page where he or she verifies that the database that has been uploaded onto the FMS is current. The next step is to initialize the position of the FMS on the POS INIT PAGE. Subsequent steps involve checking the FMS defaults, entering the flight plan data, setting parameters for departure and arrival, verifying waypoint information, setting up the display features, and entering critical, up-to-date information about the plane’s status (e.g., weight, fuel levels, cargo, passenger count) as well as the current environmental conditions (e.g., temperature, winds, etc.) Figure 3 provides a summary of the main steps involved in programming the FMS preflight.

Stage 1 (checking FMS)

1. Check STATUS	Check if database is current	FMS opens on the STATUS page or find it on the INDEX menu
2. Set POS INIT	Position Initialization: Select plane's current position	There's a POS INIT link on the STATUS page or find it on the INDEX menu
3. Check DEFAULTS	Compare defaults to dispatch release	There's a DEFAULTS link on the POS INIT page or find it on the INDEX menu

Stage 2 (entering flight plan)

1. FPLN - destination	Type in destination	Press the FPLN button
2. DEPART (+ check on FPLN)	Select SID, transition, and RWY, then verify on FPLN page	Press the **DEP ARR** button once; to verify entries, press the **FPLN** button
3. ARRIVAL	Select STARS and transition, then verify on FPLN page	Press the DEP ARR button twice ; to verify entries, press the FPLN button
4. LEGS	Verify waypoints; check cruising alts and speeds	Press the LEGS button
5. FPLN - copy	Copy the active flight plan to secondary flight plan	Press the FPLN button

Stage 3 (entering data)

1. PERF INIT	Performance initialization part 1: enter alt, fuel, wind, time	There's a PERF INIT link on the FPLN page, or on the PERF menu
2. MFD MENU	Set up the MFD Display	Press the MFD MENU button
3. PERF – THRUST LIMITS	Enter the outside air temperature	Press PERF then select THRUST LIMIT from the menu
4. PERF INIT	Performance initialization part 2: enter passenger and cargo info	There's a PERF INIT link on the THRUST LIMIT page, or on the PERF menu

Figure 3. A summary of the steps involved in programming the FMS preflight.

Like any computer, operation of the FMS by inexperienced users (students) is fraught with problems. Our own observations of student use of the FMS during graduation-level simulator sessions revealed that despite considerable training, there were still areas where performance could be improved. Chief among these is the excessive time required for (most) students to enter the desired FMS data before takeoff can occur. Many training sessions were delayed, sometimes substantially, while students hunted for the right mode button, and once inside the mode, for the right display line, to find desired information to enter, change, or confirm. Besides the lack of data entry efficiency, other chronic problems involve failing to enter one or more necessary data items, which will cause the FMS computer to operate on either faulty or missing information. Another problem is students' inability to diagnose or troubleshoot problems with the system when it fails to operate as expected. For example, failing to enter the EXEC key after each operation (an unnatural action after working on PCs), causes the system to go idle until that oversight is corrected.

The results of the CTA for this task are shown in Table 1. Note that this breakdown has been represented for the steps needed to work through any mode (five were discussed above), in which the component subtasks (and associated cognitive functions) would be performed to enter the FMS data associated with each mode button.

Table 1. Subtask and Cognitive Function Decomposition for the FMS Programming Task.

Component Subtask	**Description**	**Underlying Cognitive Function**
Knowledge of button position	The relative location of the buttons within the J-pattern, and outside the pattern, to be visually distinguished from other keys on the FMS.	Pattern Recognition
Knowledge of button functionality	This involves knowledge of the information items on each sub-menu page associated with each mode button.	Long Term Memory Access
Knowledge of button-pushing sequence	The sequence of button pushes required to enter, modify, and/or confirm various data items. It includes knowing how to sequence button pushes between line select keys, scratch pad alphanumeric keys, menu paging, and execute operations.	Procedure Execution

Table 1. (Continued)
Subtask and Cognitive Function Decomposition for the FMS Programming Task.

Component Subtask	Description	Underlying Cognitive Function
Psychomotor-manual button activation	The ability to depress the buttons with enough force to promote actuation but not so hard that data entry is unduly slowed down.	Fine Psychomotor Control
Visual translation (paper to data entry)	Rapid switching of one's gaze between the paper copy of to-be-entered data items and the FMS display without "losing one's place" or getting the two (FMS, paper copy) out of synch..	Visual Attention Attention Management
Working memory	Loading the mental representation of the to-be-entered data items in working memory until the corresponding information fields are located on the FMS display.	Working Memory Access
Keyboard entry via scratchpad	The physical act of using the FMS alphanumeric keypad to enter each data item into the scratchpad display at the bottom of the display screen.	Fine Psychomotor Control
Checking for execution	Visually monitoring the FMS display to ensure that the data item was entered into the system and then "executed" so that it makes contact with the program logic.	Visual Monitoring
Mode awareness	Keeping track of which FMS mode function (FPLAN, PERF, LEGS, etc.) is being worked on.	System Perspective
Checking for consistency	A mental and visual check that the displayed information in the FMS is consistent with the desired flight. For example, entering a 2700 instead of 270 for airspeed will make the system behave inconsistently when other operations are performed. The pilot must constantly be looking at the displayed FMS information to ensure that "out of bounds" data have not been erroneously entered.	Visual Monitoring Judgment
Checking for completeness	A mental and visual check that all data items for a given category (e.g., winds, fuel, flight legs) have been entered before moving on to the next.	Visual Monitoring

Task 2: Sequencing the FMS. Pilots are said to "sequence" the FMS when they have to access their planned route while airborne in order to make some change. These changes will be dictated by external contingencies, such as being "vectored" by air traffic control (ATC) to a given navaid in order to avoid congestion or avert weather. Other sequencing events might involve having to enter a new navaid after shooting a missed approach (e.g., unable to land due to poor visibility), changing flight altitude to achieve a more stable flight, or perhaps going to a "hold point" under ATC direction until congestion around the airfield is cleared up. In short, there are any number of possible reasons why a pilot will have to get into the "guts" of his/her flight plan and make a modification in response to some external demand. Needless to say, these demands are often not predictable, they must be responded to very quickly, and the pilot will be engaged in many other activities (e.g., radio communication, in-cockpit communications, talking to passengers) while having to do this.

The particular FMS function key(s) to be actuated (e.g., FPLAN, LEGS, RADIO) will depend on the external demand that has been placed. In some cases, multiple FMS functions will have to be accessed in order to make the necessary changes. Student pilots have problems with sequencing the FMS because this level of system interaction requires considerable knowledge of how the underlying functionality works and they must be able to recognize discrepancies in data display fairly quickly. Thus, keyboard entry must be made quickly

and accurately. The CTA decomposition of this task into component subtask and underlying cognitive functions is described in Table 2.

Table 2. Subtask and Cognitive Function Decomposition for the FMS Sequencing Task.

Component Subtask	Description	Underlying Cognitive Function
Mode awareness	The pilot must know which sensor (e.g., INS, GPS, VOR/DME) is controlling aircraft movement based on pattern of displayed data in FMS CDU.	Pattern Recognition
Spatial awareness	The pilot must know where the aircraft's flight trajectory (in 3 dimensions) is taking them relative to planned route.	Spatial Awareness
Knowledge of button position	The pilot must quickly access a given mode function soon after an external demand to change flight parameters is received.	Pattern Recognition
Knowledge of button functionality	This involves knowledge of the information items on the various sub-menu pages associated with each mode button.	Long Term Memory Access
Spatial position awareness	The pilot must know how the aircraft's current position in six axes (x, y, z, pitch, roll, yaw) relates to the location of external demand (e.g., weather cell, airfield congestion, other aircraft).	Spatial Awareness Spatial Orientation
Working memory (loading)	The pilot must put the requested information (e.g., change waypoint, increase altitude) into working memory long enough to make changes in the FMS.	Working Memory Access
Working memory (search)	The pilot must scan the contents of working memory quickly to find the needed information once in the proper FMS mode.	Working Memory Access
Mode awareness	The pilot must maintain awareness of the FMS mode he/she is in while modifying data, particularly when digging into layers of sub-menus for a given mode.	System Perspective
Checking for consistency	A mental and visual check that all just-entered data items do not conflict with other information in the FMS system (e.g., if altitude is reduced, will fuel flow still be sufficient to have proper fuel reserve upon landing?).	Visual Monitoring Judgment
Checking for completeness	A mental and visual check that all data items needed to respond to the request have been entered into the FMS.	Visual Monitoring

Task 3: Extracting to-be-entered data from the dispatch release form. Before the FMS is even operated, pilots must obtain the to-be-entered flight data from a tabular/text document from dispatch. This document contains a variety of information, for all flights, that indicates weather, flight conditions, runway variables, and a host of other data needed to program the FMS. Pilots must review these reports, quickly, to extract the data needed for *their* particular flight. Besides the environmental conditions, the dispatch release form also specifies the route they are to fly (as represented by a series of named navigation aids, navaids), number of passengers, and cargo weight. Pilots are responsible for identifying the information they need from the form, copying that information down somewhere, and then entering it correctly into the FMS once inside the cockpit. Early in training, ASU-Mesa instructors simplify this task for the student pilots by simply giving them the table of FMS data to-be-entered.

The relationship between the complex, poorly formatted dispatch release form and the table of to-be-entered FMS data is partially depicted in Figure 4. The red boxes on the dispatch release form (left side)

correspond to FMS data that must be identified and extracted. A table of *some* of the necessary data items (the complete list is longer) is shown to the right of the figure. Besides having to search through a morass of hard-to-read data, the pilot must ensure that the data they have extracted is correct (there can be errors in the dispatch release form, but it is the pilot's job to catch and correct those errors). Key data items include the flight plan (route), fuel, expected winds, and speed. Note that this is a partial list; on a given flight, approximately 20 data items must be extracted from the dispatch release form and entered into the FMS.

In Table 3, we have decomposed this extraction task into six component subtasks. The breakdown of subtasks includes a mix of behavioral and cognitive elements, although all are potentially observable (measurable). A description of how those subtasks are performed in the context of this task is provided in the middle column of the table. The corresponding cognitive functions, taken from the taxonomy in Chapter 3, are listed in the right-most column. This format is representative of all the subtask/cognitive function decomposition tables that appear in this section.

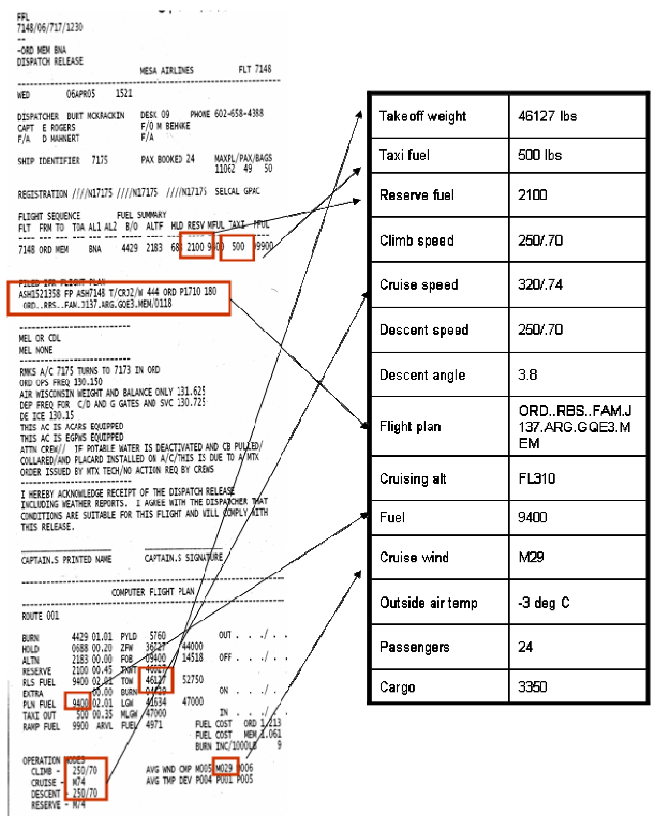


Figure 4. To-be-entered FMS data items in the dispatch release form.

Table 3. Subtask and Cognitive Function Decomposition for the Extracting FMS Data Task.

Component Subtask	Description	Underlying Cognitive Function
Visual search	Visual scan through columns and rows of data items in the release form to find categories relevant to that needed by the FMS.	Visual Attention Attention Management
Visual discrimination	Differentiate between items within a common category (e.g., winds, speed), some of which don't apply to the pilot's particular flight.	Pattern Recognition

Table 3. (Continued)
Subtask and Cognitive Function Decomposition for the Extracting FMS Data Task.

Component Subtask	Description	Underlying Cognitive Function
Load working memory	In searching the dispatch release form, only some of the items (out of ~20) will be put into working memory at one time, which will then guide the visual search of the dispatch release form. As items are found, yet-to-be discovered categories of data items must be loaded into working memory.	Working Memory Access
Knowledge of relevant information	From long term memory, the pilot must remember the data items needed by the FMS, the format (e.g., units of speed, wind) they are in, and how those items relate to the flight he/she is about to plan.	Long Term Memory Access
Search working memory	As the dispatch release form is visually scanned, the pilot continually checks the contents of working memory to see if any of his/her needed (by the FMS) items is on the form. There is a continuous interplay of visual search and working memory search to guide data item matches; as each item is matched, it is extracted and taken from working memory.	Working Memory Access
Check for completeness	A visual search of the text-based release form to find any needed FMS data items that may be missing. Some are hard to find and the pilot can't forget to find them and extract them from the form before beginning FMS data entry.	Resource Management Visual Monitoring

Task 4: Recognizing and remediating errors and discrepancies in the FMS. The fourth task that student pilots must perform with the FMS is to recognize when the FMS is operating with discrepancies. This is most often manifested when the FMS encounters a "discontinuity," that is, when there is a gap between one or more waypoints in the flight plan so that there is no longer one, continuous route. The underlying reasons for discontinuities are several and the bases for them are fairly technical. Some discontinuities are actually not errors in programming, but are a natural part of the flight plan. This occurs when there is no preplanned navaid near the destination airfield, where it is expected that ATC will vector the aircraft near the airfield to some holding point or to some navaid that cannot be identified until the aircraft actually is on approach. In other cases, the discontinuity has arisen because there is reason to treat the route as several discontinuous or disconnected segments. This segmentation happens when the nature of the maneuvers near a waypoint (e.g., fly-over vs. turn) cannot be anticipated until the time of flight. But regardless of the underlying reason for a discontinuity, it is the pilot's responsibility to (1) recognize when the FMS is about to enter a "discontinuity" state and then (2) take the proper corrective action.

Discontinuities will appear as yellow-text (FPLN DISCONTINUITY) in the message line of the FMS. Once the message appears, the pilot has 2 min. until the flight plan will have sequenced past the last waypoint prior to a flight plan discontinuity. As time passes without one of the pilots noticing that the discontinuity message is being displayed, they have that must less time until the FMS reaches a state when it is no longer operating on active data. Yellow is used only to code cautions and error messages in the FMS (other colors are green, magenta, white, blue), so simply seeing the color yellow is a good stimulus to action. Actions the pilot might take include accessing the LEGS page and inserting a new waypoint, deleting the waypoint (or navaid) that makes the route discontinuous, or some other action.

Besides discontinuities, the FMS might display yellow caution messages when the aircraft's position was initialized incorrectly, the aircraft's vertical flight envelope will violate constraints (e.g., altitude constraints to clear an upcoming obstacle), or there is a disagreement among the aircraft's navigation sensors, to name but a few. The nature of the caution or error message will dictate the appropriate response the flight crew is

to take. Typical student problems with this task include failing to detect the messages in sufficient time to take corrective action, becoming confused about which particular problem the FMS is having, and failing to take the proper corrective action.

Table 4. Subtask and Cognitive Function Decomposition for the Recognizing FMS Errors Task.

Component Subtask	Description	Underlying Cognitive Function
Knowledge of button functionality	Knowing which mode functions are going to be needed for a given type of error message or caution.	Long Term Memory Access
Mode awareness	The pilot must know when the FMS is in, or about to be in, an error state or is displaying a caution mode.	Detection Pattern Recognition
Spatial position awareness	The pilot must know how the aircraft's current position in six axes (x, y, z, pitch, roll, yaw) relates to the location of external demand (e.g., weather cell, airfield congestion, other aircraft).	Spatial Awareness Spatial Orientation
Knowledge of system coding	Knowing the meanings associated with each of the FMS' five color codes, with Yellow signifying caution or error messages. Also knowing the meanings of the text associated with different caution and error messages.	Long Term Memory Access
Knowledge of button position	The pilot must quickly access a given mode function soon after an error/caution message is detected.	Pattern Recognition
Knowledge of system logic	The pilot must know which data items can be entered, which functions can be accessed, and which modifications can be made and still be consistent with the system's underlying logic of operation.	Long Term Memory Access Critical Thinking
Translate spatial picture to system picture (mental model)	Mental transformation that the spatial array of entities in the real world (e.g., weather cells, aircraft, navaids) matches that being depicted in the FMS display.	Visualization
Checking for consistency	A mental and visual check that just-entered data items do not conflict with other information in the FMS system (e.g., that deleting a waypoint to eliminate a discontinuity won't create a discontinuity if the alternate airfield has to be flown to).	Visual Monitoring Judgment
Checking for completeness	A mental and visual check that all data items needed to correct the error have been entered into the FMS.	Visual Monitoring

Competency 2: Profile-Based Callouts

Overview

The competency that we are calling Profile-Based Callouts (PBCs) is an integral aspect of the safe and efficient operation of multi-pilot aircraft. To appreciate this competency, we first describe several overarching cockpit tasks that provide the context within which this activity occurs.

We start with the notion of flight *profiles*, which can be viewed as a graphic representation of some maneuver that the aircraft makes at a certain point during the flight. Each airline defines their own set of prescribed profiles, although many are common across airlines. Mesa Air has 16 distinct profiles, some which, like normal takeoff, ILS approach, and normal landing, occur on every flight. Others, such as rejected takeoff or single engine landing, occur only once in awhile or during emergency situations. Associated with each flight profile is a procedure, which is a highly detailed step-by-step explanation of how the maneuver (e.g., single engine approach) is to be accomplished. The FAA has prescribed guidelines for procedures, which are then adapted by each airline to become one of their standard operating procedures or SOPs (Loukopoulos, Dismukes, & Barshi, 2001).

Each procedure may be broken down into tasks and even further into task steps. These steps must be learned to such proficiency by pilots that they become automatic; hence, a degree of overlearning is expected from most procedures. A major element of any procedure is *flows*, which is an organized systematic rehearsed method of scanning the instrument panel (IP) that is interspersed with task actions. These flows give the pilot a systematic and efficient visual strategy to examine the IP from top to bottom and from left to right, so there are no overlaps or loops. A flow is referred to as a memory item since it must be performed immediately and from memory once some triggering event has occurred. They are done silently by the designated crew member (CM).

For example, there is a flow procedure that must be performed after the engines have started. The triggering event is that the engines are now running and the plane is getting ready to taxi. In the Captain's flow, he/she will systematically scan the generator electrical, hydraulics, rudder, and rear wheel steering to ensure they are operating normally. Simultaneously, the first officer (FO) will be scanning ignition, packs, anti-ice, and probes to make sure they are working. Typically, the flow is punctuated with a checklist, consisting of a subset of the important ("killer") items from the flows. This subset is addressed in the checklist, which is written down using a company-prescribed format, with one CM reading off the items and the other CM confirming verbally that the item has been checked, performed, or verified.

The flows and checklists allow the CMs to form habitual associations among tasks, such that the presence of switch settings and panel information serves to cue the pilot what to do and/or look at next. Through training and extensive practice, these associations are assumed to "solidify into habitual, automatic action sequences that can theoretically be relied upon … "(Loukopoulos, et al, 2001). In essence, these associations are the glue to hold the flight operation together when workload is increased due to interruptions, weather, traffic, or some in-flight emergency.

Overlaid on this operating milieu are *callouts*, which are a verbal method to ensure coordination between CMs. Part procedure and part technique, callouts entail a scripted, choreographed interchange of visual scanning, task actions, and verbal challenges and replies that are initiated once the aircraft reaches one of its pre-designated profile situations. At that point, each pilot begins performing their own flow, consisting of scanning the IP, checking switch settings, and taking actual physical actions. At predetermined times in these flows, pilots will make callouts to each other, to ensure that that pilot has performed the required actions or to inform the pilot of his/her own actions.

As an example, consider the engine failure after V1 (the speed at which the aircraft can no longer abort a takeoff) flow shown in Figure 5. It is highly scripted, where the specific actions required of each pilot are designated by text not in quotes, and the specific verbal utterances are in quotes. The graphic also shows that the timing of the flow and callout initiation in the profile is important as is the sequence. Because of the intricacies in synchronizing these actions/verbalizations, pilots must work together during training to get their timing and cadence down so they don't "step on each other's comms." Also, the interchanges should be smooth and not jerky, where one pilot is waiting for another to say something (just like one actor waiting for

another to say his lines). Our observations, confirmed by discussions with SMEs, is that students never practice these PBCs enough, on their own. When it comes time for simulator training, there can be long delays in the session while students have to “remediate their callouts and flows” before proceeding with actual aircraft operation. Not surprisingly, crew coordination is a very important component of this competency.

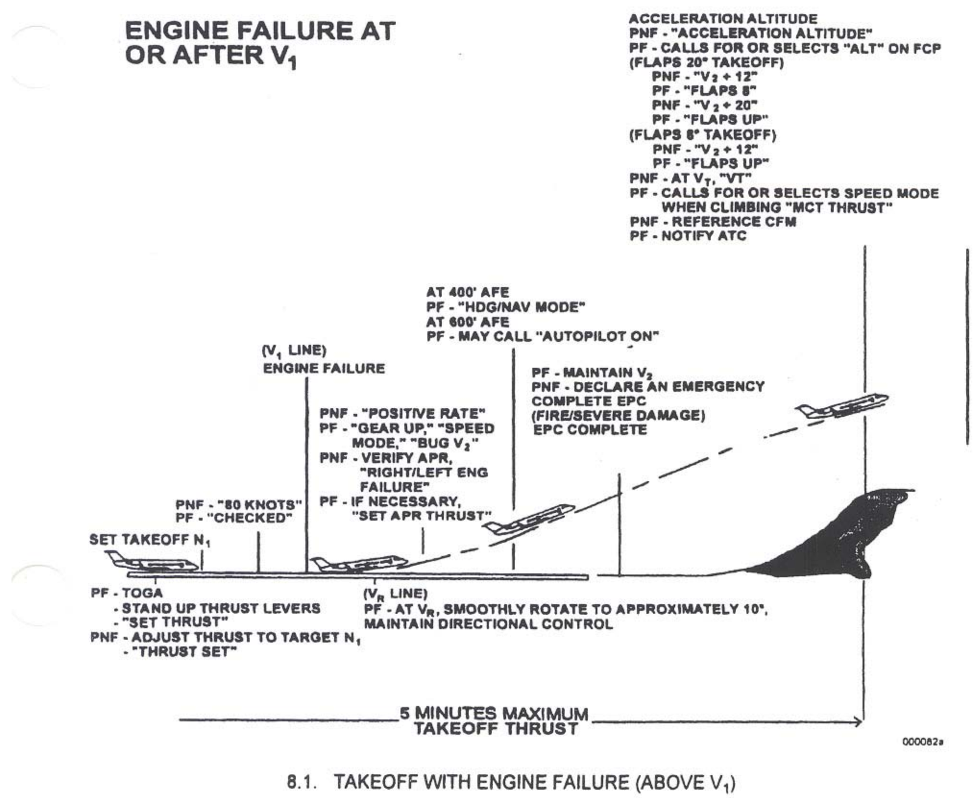


Figure 5. Example callout profile for Engine Failure after V_1.

Cognitive Task Analysis

As with the FMS, we performed a mid-level cognitive task analysis (CTA) on the PBC competency. The CTA revealed that PBCs can be divided into four tasks: (1) detecting the profile situation, (2) initiating the flow, (3) synchronizing their callouts with the flows required for that profile, and (4) coordinating the flows and callouts with the other CM. Our recent observations at ASU-Mesa and in-depth interviews with expert pilots indicated that it is important to focus on the choreographed interplay of pilot actions and verbalizations, as this underlies much of what makes a given flight successful. As well, this part of the larger flow of airline operations has great potential for improvement through innovations in training, such as GBT, where new possibilities to induce practice could yield immense dividends. Below, we decompose each task into its component cognitive tasks.

Task 1. Detect Profile Situation. The overarching trigger event for this competency is detecting that the aircraft has reached or is about to reach a point where a profile is encountered. As noted above, Mesa Airlines and ASU-Mesa have identified 16 situations/profiles where callout-flow sequences are used. For this task and the other three PBC tasks, our CTA breaks down the task into component subtasks, an activity description, and associated cognitive functions. Table 5 displays the results of the CTA for this first task, Detect Profile Situation.

Table 5. Subtask and Cognitive Function Decomposition for the Detect Profile Task.

Component Subtask	Description	Underlying Cognitive Function
Maintain spatial position awareness	The pilot must know how the aircraft's current position in six axes (x, y, z, pitch, roll, yaw) relates to the onset of each of the profiles.	Spatial Awareness Spatial Orientation
Access long term memory	The pilot must have the onset characteristics of all 16 profiles represented in long memory.	Long Term Memory Access
Load working memory	Key characteristics from relevant profiles should be put into working memory in some useful representational form.	Working Memory Access
Search working memory	Candidate profile onset characteristics should be continually searched in working memory as aircraft continues its flight.	Working Memory Access
Visualize flight trajectory	The pilot must mentally image where the aircraft will be, in 3 dimensions, x seconds from the current time and position.	Visualization Situation Awareness
Detection	The pilot must detect—using visual, auditory, or kinesthetic means—that the triggering stimulus for a profile is present.	Detection
Visually scan instruments to confirm	The pilot must recheck instrument to ensure that the triggering event for the profile is stable, and should be acted upon.	Visual Monitoring

***Task 2. Initiate Flow*.** Once pilots have detected they are in a "profile situation," the next task is to initiate the sequence of activities (flows, task actions, callouts) that comprise that profile. This initiation must occur at the proper time in the profile; it cannot either be too early or too late. Also, while both pilots have their own flows to perform, some profiles require that a particular crew member initiate the sequence. For example, in the Rejected Takeoff profile in Figure 6, the sequence begins with the pilot flying (PF) manipulating the thrust levers and then saying "thrust set." If pilots fail to start their flows and callouts on time (much like an actor "missing their mark"), the entire sequence is thrown off, and it will put the CMs in an awkward hurry-up mode that will severely degrade their performance throughout the rest of the profile. Integrated, synchronized team practice is needed on many of these profiles to ensure that the flows are initiated at the proper time. The results of the CTA for this second PBC task are depicted in Table 6.

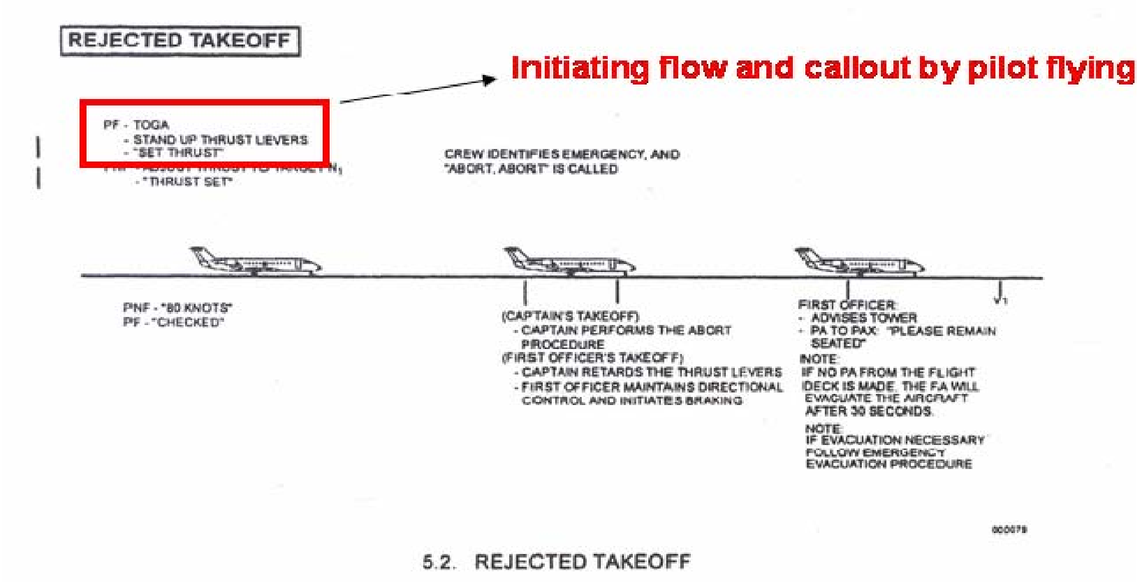


Figure 6. Callouts and flow for the rejected takeoff profile.

Table 6. Subtask and Cognitive Function Decomposition for the Initiate Flow Task.

Component Subtask	**Description**	**Underlying Cognitive Function**
Visual sweep across instrument panel and OTW	Vigilant monitoring of displays (heads down) and out the window (heads up) to look for cues indicating that the callout or flow should be initiated.	Visual Monitoring
Time start of flow to proper position in profile	Recognize cues in the profile (visual, auditory), both within the cockpit and OTW, that correspond to known points in the profile.	Pattern Recognition
Synchronize running the flow with relative time in profile	Have an "internal clock" that keeps track of the time that has elapsed between flow events, and then establish a common starting point for the flow sequence with the passage of time in the profile.	Time Estimation Timed Response
Access long term memory	Retrieve from long term memory the representations of cues to guide the conduct of flow tasks associated with the particular profile.	Long Term Memory Access
Load working memory	Represent the cues for the immediate flow task in short term memory.	Working Memory Access
Perform flow tasks IAW pre-learned organizational scheme	The flow tasks (switch settings, panel scanning, information confirmation, etc.) are well-learned actions that should adhere to established company airline procedures.	Procedure Execution
Visual check that other CM is in his/her flow	The pilot needs to periodically check over to the other crew member to ensure that he/she is engaged in their own flow, and not waiting for some action by the pilot.	Crew Coordination Visual Monitoring
Visual check that own-flow tasks are completed	Scan the area of the cockpit (controls and displays) to ensure that no scanning or flow actions have been missed from the prescribed procedural sequence.	Visual Monitoring

Task *3. Synchronize Callout with Profile and Flow*. Pilots make callouts for several reasons. On the one hand, the pilots will issue statements of confirmation, such as "checked," in response to the other CM's verbalization of the aircraft's present speed (see Figure 6). Other times, callouts are given as commands, where one pilot might say "set thrust" as a cue for the other pilot to perform that physical action. Like any script, the callouts are to be spoken exactly as written in the procedure, in the proper order, and at the proper time. Because most of the procedural actions must be done quickly and in a precise order, there is neither room for improvisation nor any amplifying verbalization. Most importantly, the pilots' verbal callouts must be synchronized with their flows, so that each pilot is "looking-thinking-talking-doing" in the right proportion, at the right time. Demands on attention management are very high, and it takes considerable practice for the beginning pilot to learn how to tradeoff visual scanning, talking, and physical actions within each profile. The results of the CTA for Task 3 are summarized in Table 7.

Table 7. Subtask and Cognitive Function Decomposition for the Synchronize Callout Task.

Component Subtask	**Description**	**Underlying Cognitive Function**
Access long term memory	Remember cues in the profile that serve to stimulate a verbal callout and remember the precise wording for each callout.	Long Term Memory Access
Load working memory	Represent the cues for the immediate callout in short term memory.	Working Memory Access
Time verbal response to synch with flow tasks	Maintain proper balance and order of verbalizing the scripted callouts to reflect the stated intent in the profile (i.e., a command, confirmation, challenge, response).	Time Estimation Gross Psychomotor Control
Maintain cadence of verbal response	Ensure that the time between utterances is correct and consistent with what is needed for the profile so that own-flow tasks are not disrupted.	Attention Management
Visual check that other CM is in his/her flow	The pilot needs to periodically check over to other crew member to ensure that he/she is engaged in their own flow, and not waiting for some action by the pilot.	Crew Coordination Visual Monitoring
Visual check that own-flow tasks are completed	Scan the area of the cockpit (controls and displays) to ensure that no scanning or flow actions have been missed from the prescribed procedural sequence.	Visual Monitoring

***Task 4. Coordinate Flow and Callouts with other Crewmember*.** Finally, note that the callout-flow routines are a social activity, where each CM must not only focus on his/her own responsibilities, but be available to help the other CM with theirs. This is the essence of crew coordination, where each pilot's verbal callouts stimulate, command, challenge, or confirm actions from the other pilot. Like a choreographed dance, the two pilots' own flows and callouts must be properly in balance, else the stated intent of the callout (e.g., confirmation or a challenge) will not be achieved. This balance requires that pilots rehearse these profiles both with individuals they actually will fly with, as well as with instructors from whom they may obtain feedback on proper cadence, intonation, and timing.

There is no simulator equipment specifically designed for this instruction, so pilots are expected to use physical mockups, part-task trainers, and old-fashioned chair flying to create a physical context in which flows may be integrated with scripted callouts. There is no tangible product (or artifact) that results from these practice sessions, so it is difficult to monitor how well or how much students are engaging in the necessary practice. It appears to us that most students do not avail themselves of the opportunity to practice these procedural actions, either with another student or on their own. The results of the CTA for Task 4 are shown in Table 8.

Table 8. Subtask and Cognitive Function Decomposition for the Coordinate Flow and Callouts Task.

Component Subtask	Description	Underlying Cognitive Function
Time cadence of verbal response to match other CM's task flow	The pilot must gauge the beginning, duration, and intervals of his/her verbal callouts to coincide with the flow tasks being performed by the other pilot.	Time Estimation Timed Response Attention Management
Verbally respond to other CM's challenge	The pilot must maintain in memory the required verbal response for each challenge the other crew member might make during the profile.	Long Term Memory Access
Perform own-flow tasks to synch with other CM's callouts	Pilots must divide their attention so they can continue to visually scan instruments and perform actions required in their flow while retaining auditory attention to respond to other crew member's callouts.	Procedure Execution Attention Management
Perform own-flow tasks to synch with other CM's flow tasks	Pilots must adhere to the procedure sequence learned in that profile so their flow tasks do not compete with, disrupt, or interfere with tasks being performed by the other crew member.	Procedure Execution Shared Understanding
Visually scan instruments to maintain flow	Visually scan the displays in a prescribed sequence and time interval so the pilot's own flows can be accomplished accurately and quickly.	Visual Monitoring Visual Attention
Visual check that other CM has maintained their flow	The pilot needs to periodically check over to other crew member to ensure that he/she is engaged in their own flow, and not waiting for some action by the pilot.	Crew Coordination Visual Monitoring

The Taxonomy of Cognitive Functions

After conducting the CTA of the FMS and PCB training environments, the next step was to integrate the findings into a taxonomy of key cognitive functions and competencies that are commonly found in these types of training environments. Table 9 lists 26 cognitive functions that were identified based on the CTA and a review of standard literature sources in cognition, focusing on aviation aspects, and vetting the list against prominent learning theories in education (Bransford et al., 2006). These cognitive functions along with the serious game element taxonomy described in Chapter 2, form the foundation for the cross-walk between gaming elements and the cognitive functions that underlie the component subtasks in our two selected competencies. The left column of Table 9 lists the cognitive function; the right column provides a brief definition of how that concept will be used in the present context.

Please note that this taxonomy is by no means intended to be a comprehensive account of all the cognitive functions that might reasonable be expected to underlie complex skill performance. Rather, it was specifically designed to support the aviation competencies examined in this Phase I. However, our taxonomy was certainly developed with an eye toward eventual expansion to other competencies and application domains. It should be viewed as a "work in progress" and further development – in the form of refinement, redefinition, further identifications, etc. – will occur in Phase II. For convenience of presentation, the cognitive functions are listed in alphabetical order.

Table 9. Cognitive Functions for Use in the Gaming Element – Aviation Competency Crosswalk.

Cognitive Function	Working Definition
Attention Management	Controlling or dividing attention, across modalities, and across various aspects of the cockpit, including OTW and instrument panels
Crew Coordination	Working in concert with another crew member (CM), either through verbal exchange, backing up (checking that other CM's responses are correct), or working separately on shared tasks
Critical Thinking	Reasoned thinking that reflects on the outcome of just-completed rapid, automatic processing.
Decision Making	Selecting one option from among several
Detection	Making a simple response once sufficient sensory information (visual, auditory) has been obtained
Fine Psychomotor Control	Making frequent, small input corrections based on a combination of visual and kinesthetic feedback
Gross Psychomotor Control	Making infrequent, large input corrections based on a combination of visual and kinesthetic feedback
Induction	Deriving or discovering some rule based on the presence of several example facts
Judgment	Selecting course of action based on intuition, "feel," and hard-to-articulate prior experience
Long Term Memory Access	Retrieving learned procedures, rules, or information for use in current or upcoming tasks
Pattern Recognition	Accurately identifying a constellation of visual stimuli as representing some previously learned category based on perception of the entire array or on parts of the array
Procedure Execution	Engaging in some previously learned, well-rehearsed, organized sequence of actions
Resource Management	Allocating one's mental, sensory, and physical resources in order to minimize immediate or long-term workload and improve overall task performance
Shared Understanding	Working in synch with other CMs through the process of mutual monitoring, periodic communication, and focused feedback.
Situation Awareness	Being able to project one's trajectory over time and space to predict status and position at some future point
Spatial Awareness	Recognizing one's position in 3-dimensions based on reference to the ground and projected movement through space
Spatial Orientation	Maintaining plane orientation in space relative to the horizon
System Perspective	Maintaining the big picture of where a given piece of information or event fits within the larger whole
Task Management	Being able to prioritize tasks, share tasks, delay tasks, and shed tasks in order to avoid task overload
Time Estimation	Determining passage of time based on an internalization of the sequence and flow of events rather than observing instruments

Table 9 (Continued)
Cognitive Functions for Use in the Gaming Element – Aviation Competency Crosswalk.

Cognitive Function	Working Definition
Timed Response	Engaging in prescribed sequence of responses based on estimated passage of time or perception of appropriate trigger event
Visual Attention	Maintaining a visual focus on some area of the world (OTW, display) with sufficient duration that information can be extracted and processed
Visual Estimation	Arriving at an accurate prediction of the size, number, or extent of some stimulus without actual counting or measurement
Visual Monitoring	Visual observation of display, OTW event, or other CM to confirm that some desired information, effect, or action has occurred
Visualization	Forming and maintaining a stable, accurate mental representation of some visual entity and representing one or more transformations of that entity based on future interactions
Working Memory Access	Searching the contents of working memory to find the information needed to perform the task at hand

Chapter 4: Development of Game Element – Training Competency Crosswalk

After developing the game element taxonomy and the cognitive function taxonomy, we then began work on integrating the two in a preliminary crosswalk designed to link particular game elements with different training environment needs. The crosswalk is based on findings from serious game studies, such as the ones described in Chapter 2, as well as research and theory in cognitive psychology, educational psychology, multimedia design, and instructional design. Research in these areas can inform us, for example, which game *structure* is best suited to foster procedural skill development, how *social interaction* can be structured to foster self-efficacy and improved learning outcomes, and what type of *feedback* is most appropriate in a given situation. Table 10 shows an example of how some cognitive functions found in many training environments may be linked with some key game element variables from the serious game element taxonomy outlined above. Column 1 lists the cognitive function. Column 2 lists the broad category of a game element that might be particularly relevant in that environment. In columns 3 and 4 we briefly describe a sample research study that illustrates the rationale behind the linkage.

Table 10. Preliminary Crosswalk Linking Some Cognitive Functions with Associated Gaming Elements.

Cognitive Function	Potentially Associated Game Element	Article	Summary
Attention Management	Situational Realism (type: cognitive)	Gopher, et al. (1994)	Those who played game with low physical similarity, but high cognitive similarity (e.g., competing task demands) – whether with part-task training or emphasis-change training performed better on subsequent assessments of actual flight performance than those who did not play game. Compare to Hart and Batiste study in which those who played a different game (with different demands) did not improve.
Crew Coordination	Social Interaction; Structure	Shelbilske, et al. (1992)	Those who practiced the Space Fortress game using a dyadic approach (Active Interlocked Modeling) in which players trade off controlling specific tasks in a whole task environment, did just as well on subsequent flight performance measures as those trained in an emphasis-change environment (where one player plays in a whole task environment but focuses on different tasks as different times). Players engaged in a high degree of communication and cooperation

Table 10 (Continued)
Preliminary Crosswalk Linking Some Cognitive Functions with Associated Gaming Elements.

Cognitive Function	Potentially Associated Game Element	Article	Summary
Decision Making	Cognitive, emotional, and behavioral realism (time pressure, uncertainty, stress)	Lewis & Barlow (2005)	Teams used first-person shooter game to play out four scenarios involving dynamic decision-making (involving making rapid, sequential decisions, the outcome of which affects other decisions); teams who scored higher were rated as using experience more effectively in the DM process, and having better Situational Awareness
Detection	Challenge; cognitive realism	Green & Bavilier (2003)	Non-video game players who were asked to play an action game requiring simultaneous juggling of many tasks, first person perspective, and requirements to monitor entire visual field did better on tests of visual attention than Non video game players who did not play the game.
Long Term Memory Access	Feedback (level)	Cameron & Dwyer (2005)	Game asked factual and conceptual questions about physiology of human heart. Those given elaborative feedback did better on delayed retention than those given confirmatory or no feedback – or those who received questions in a non-game format.
Pattern Recognition	Instructional Support (type)	Mayer, Mautone & Prothero (2002)	Those who received pictorial scaffolding prior to playing a game (showing how to interpret data patterns) did better on game and transfer than those given strategy tips.
Situation Awareness	Cognitive, emotional, and behavioral realism (time pressure, uncertainty, stress)	Lewis & Barlow (2005)	Teams used first-person shooter game to play out four scenarios involving dynamic decision-making (involving making rapid, sequential decisions, the outcome of which affects other decisions); teams who scored higher were rated as using experience more effectively in the DM process, and having better Situational Awareness
Task Management	Situational Realism (cognitive/emotional realism)	Morris and Shirkey, (2004); Gopher et al. (1994)	Adding pre-training, which introduced stressors to increase situational/emotional realism, resulted in improved performance on a military simulation game (factors of which were strategy development and task persistence) compared to those who played same game without the pre-game stressors.
Time Estimation	Situational Realism (cognitive/emotional realism)	Morris and Shirkey (2004)	(see above – assessment also included time management)
Visual Attention	Challenge; cognitive realism	Green & Bavilier (2003)	Non-video game players who were asked to play an action game requiring simultaneous juggling of many tasks, first person POV, and requirements to monitor entire visual field did better on tests of visual attention than NVGPs who did not play the game.
Visual Monitoring	Challenge; cognitive realism	Green & Bavilier (2003)	(see above: the game also appeared to widen players' visual fields, as measured by differences in pre and post-game perceptual test.
Visualization	Instructional Support (type)	Mayer, Mautone & Prothero (2002)	Those who received pictorial scaffolding prior to playing a game (showing how to visualize and interpret data patterns) did better on game and transfer than those given strategy tips.

The crosswalk is still in a preliminary stage of development, with more work needed in the transition period and Phase II, should funding be provided. There are three areas where continued development is needed, and each is essential to the long-term success of the project. First, the crosswalk or matrix must be linked to or "populated" with additional findings from the research and gaming literature. While we know that the amount of scientifically credible research on game variable effects is limited, there are pockets of applicable research in several related domains. These include multi-media research, the occasional study in the

business realm (e.g., organizational research, training development, business or management games), and the burgeoning field of serious game insertions in various academic and school settings. Though chaotic as a literature, these domains can be gleaned for “nuggets” of applicable research and we will do so in the early stages of Phase II and throughout the project. This will constitute further “population” of the crosswalk.

Second, we will need to perform further organizational and conceptual development so that we may turn our crosswalk, which is presently cast as a flat-structured Excel worksheet, into a true relational database. This is what we have referred to as the TARGET tool. When represented in this form, we will be able to use our key fields (such as a particular game element variable, such as challenge or feedback) as a common link to our research base noted above. But more importantly, this relational structure will allow us to solicit additional information from various sources by casting the tool as an XML object which can be distributed across the Web for feedback, input, and even trial-use. That is, we will use various forums, such as the Serious Game ListServe, as a way to solicit key field-driven queries on specific topics that can be incorporated into an ever-expanding database tool. The tagging capabilities of XML will be invaluable for compiling comments, feedback, and suggestions from a large number of recipients on a daily and weekly basis.

For example, in areas where there are little (if any) research data, we may use our XML-structured TARGET as a way to at least acquire from practitioners “best practice” information concerning how a given game element (e.g., use of high-value reference figures, such as a teacher or mentor, in a virtual or avatar form) might be employed. As well, we would use the tool to collect user feedback concerning the success or failure of existing games in various training environments. In some cases, the information gleaned from the broader field might give rise to suggested research studies that we would perform in Phase II, as we have described in Chapter 8. Besides collecting far-ranging input information, TARGET would be utilized on the output/report side by generating profile descriptions of various training environments to which our cadre of game elements would be targeted. Thus, TARGET could be exercised, like any database, to generate recommendations for how a given game feature might be incorporated into a training environment having a particular set of characteristics.

Third, as our crosswalk (Excel) and TARGET tool (XML) are reorganized and populated, we will begin to look for patterns or trends in areas where research is either missing or inconclusive. In performing the conceptual and “networking” tasks for this project it has become clear that, while many highly placed analysts and researchers in government and industry are eager to see serious games get “greater air time” in training, there is a definitive wait and see attitude until more compelling research data are collected. By having our tool specifically highlight where the gaps and inconsistencies are most prevalent, we will be in a position to either conduct or suggest targeted research to rectify those shortcomings. In this way, we should be able to advance the field of GBT at faster pace than simply doing a single parametric study at time.

In the next section, we describe how our preliminary crosswalk was utilized to lay the foundations for designing the FMS Programming Game. Though represented in far less detail, the crosswalk was also used to provide preliminary specifications for the callout procedure game as well.

Applying the Crosswalk to the FMS Training Environment

In this section, we outline our general approach for designing the FMS game – specifically, how we took the main characteristics of the FMS training environment and applied the principles underlying the preliminary “Game Element – Training Competency Crosswalk” to guide the selection of optimal game elements to incorporate into the game design. In chapter 5, we describe the FMS game in more detail, highlighting the main features of the game storyboards we created in Task 4,

Below, are some examples of how we used the crosswalk to select specific game element variations for our FMS training game. We first provide a brief synopsis, listing a major category from the game element taxonomy, provide a brief summary of a characteristic of the FMS training environment, some of the associated cognitive functions, and the game element variation suggested by the crosswalk. We then provide a more detailed description and rationale of the selection process.

Game Element 1: Game Structure and Instructional Support

⬩ *Training Environment Characteristics:* Students need to (1) have a “big picture” understanding of the FMS AND (2) must master basic component skills

⬩ *Associated Cognitive functions:* long term memory access, procedure execution, attention management and resource management, as well as fine motor control, pattern recognition, detection

⬩ *Recommended Game Element Variation:* Use a combination of a main game, with a scaffolded, whole-task structure, and an embedded game consisting of a set of gamelets that provide rapid, component-task training.

In programming the FMS, students need to have a “big picture” understanding of how to interact with the FMS in a real-world environment (i.e., the cockpit of a commercial jet). This includes applying the appropriate sequence of steps to take while programming the FMS in preparation for takeoff, as well as handling the pressures of performing the task under tight time constraints, with frequent interruptions. It also involves knowing what data to update and how to make the necessary changes as new information comes in. Knowing what information is relevant during a specific phase of flight, and how to obtain it (e.g., from a dispatch release or from ATIS) is also important Programming the FMS also involves detecting errors and handling routine and non-routine problems, which, in turn, requires an understanding how the FMS interacts with other equipment to affect performance of the aircraft.

A common problem for example, is forgetting the sequence of steps necessary to insure that all information, such as the coordinates, flight plan, current data on fuel, temperature, weight, etc., has been correctly entered and/or verified. Leaving out a step can have serious consequences, as can taking too much time to get to the next step, Therefore, it was initially decided that when it came to how we would **structure** the game, rather than just take a traditional part-task approach (where the students initially receive intensive training on isolated tasks before integrating them into a whole), it was more appropriate to use a scaffolded, whole-task approach. With this approach, students are exposed to the whole task throughout the training, but are only required to perform some parts of the task while the other parts are initially performed by someone else (in the case of our game, a virtual captain) who explains what he or she is doing. This way, students receive repeated experiences with the entire process, but are not cognitively overwhelmed. As students progress through the various levels, becoming more familiar with the environment and more adept at completing the tasks, they take on increasing responsibilities until they are able to perform the task to a criterion standard with little or no assistance. Research in both general instructional design (Mayer, 2005; Sweller 1999) and game based training (Gopher, Weil & Bareket, 1994), suggest that this type of scaffolding and cognitive apprenticeship not only helps learners understand the steps and see how the whole process works, it also better prepares students for the cognitive demands of the actual training environment.

On the other hand, we realized that many of the component tasks involved in working with the FMS, such as using the appropriate buttons and line select keys to rapidly navigate to specific pages within the FMS, involve procedural motor and visual skills. Developing these skills requires a great deal of repetitive practice before students can reach a level where performance is automated (Anderson, 1993). Therefore, in order to provide the type of practice needed, we chose to also include a set of rapid, part-task training tasks – in the form of arcade-type games – that the players access between levels of the whole-task game. These games would not only provide players with the repetitive practice needed to automate the component skills and provide a chance for players reinforce some of the skills needed in the whole-task game, they also provide a variation in the pacing of the game

Game Element 2: Situational Realism

⬩ *Training Environment Characteristics:* Skills will be applied directly to work environment

⬩ *Associated Cognitive functions:* system perspective, long term memory access, procedure execution, attention management and resource management, as well as fine motor control, pattern recognition, detection,.

⬩ *Recommended Game Element Variation:* The storylines, interface, and functionality employed in the game as well as the behaviors required of players should employ situational, behavioral, and cognitive realism

Situational realism refers to how well the physical aspects of the game, as well as the cognitive, emotional, and behavioral aspects, correspond to the real-world environment in which the target skills are actually performed. Programming the FMS is something that trainees will be carrying out not only in the ASU simulators, but very shortly, in their professional careers as pilots. Because the skills and procedures students learn during training will be applied almost directly to actual work environments, it is especially important that students have the opportunity to experience, as much as possible, how members of their profession think, behave, and solve problems. This is referred to in the cognitive literature as distributed authentic professionalism and situated learning (Aldrich, 2005; Gee, 2005; Mayer, 2003). Research on general and specific intelligence has repeatedly shown that embedding skills in the context in which they will be applied is more effective than attempting to create a program that trains “general skills” (Mayer, 2003). Therefore, rather than create a generic game interface, we chose, in The FMS Programming Game, to create a simple but visually and auditorily realistic interface that replicates much of the functional experience of a cockpit-resident FMS. We also provided a “virtual captain,” as well as other equipment that one might use in an actual cockpit environment. In addition, we created fairly realistic storylines in which the player assumes the role of the First Officer who is mentored by a virtual captain (played by the computer), where the latter interacts with the FMS game player much like a real captain might do. Players are required to behave as they would if they were actual First Officers in charge of programming the FMS. The storyline of the game attempts to make the game relevant and provide players with a reason for knowing evident rudimentary information about the FMS. In addition, the types of feedback the game provides attempts to replicate what might actually happen in a real situation, such as running short on time, using up fuel, receiving commendations or disapproval from the captain, having to return to the gate, dealing with impatient passengers, and so on – all realistic consequences for success or failure. Incorporating this type of context-rich role-playing, in which players develop and refine their strategies by applying them in a variety of situations, is key to fostering meaningful understanding and improved retention and transfer, particularly for this type of training environment (Aldrich, 2005; Mayer, 2003).

Although the game attempts to maintain as much authentic realism as possible, the FMS Programming Game still includes elements of “fantasy” or suspended realism. In the real world, a novice would not be given the responsibilities that the players in this game are given, nor would players be able to see the consequences of their actions so quickly (in some levels of the game, where students are allowed to make errors without the system giving them immediate feedback, they may see the plane may veer off course, or even crash). Other features, such as including a side game panel with feedback, timers, an overview of the taxiway, as well as other cues to inform players about how close they are to reaching their goal, are not something that would be found in an actual cockpit or cockpit simulator. Yet they are still relevant to the theme of the game, and reinforce some of the “natural” consequences for success and failure, such as holding up other planes on the taxi way if one takes too long to program the FMS.

Game Element 3: Challenge

⬩ *Training Environment Characteristics:* The overall tasks are complex and composed of several procedural sub-tasks with the higher-level tasks building on the lower level tasks. Many of the subtasks are not inherently interesting, and students learn the material at different rates. Some of the main tasks in FMS programming involve routine and non-routine problem-solving

⬩ *Associated Cognitive functions:* attention management, critical thinking, induction, etc.

⬩ *Recommended Game Element Variation:* Incorporate multiple challenge levels, uncertainty,

Another game element that we focused on was how to make the game **challenging.** As noted above, many of the tasks involved in working with the FMS – in particular, the data entry and verification – are procedural and thus require a great deal of repetitive practice. While automation is a worthy goal, it is difficult to achieve since many of these lower level tasks are inherently boring; therefore, in order to engage players and encourage them to practice enough to master these skills, the game must build in motivating challenges. It also needs to accommodate players with potentially varying abilities, some of whom may acquire the skills quite rapidly, others of whom may need more practice.

As alluded to above, The FMS Programming Game is designed around the idea of graduated complexity, consisting of several levels to which players only advance when they reach some criterion level of mastery

in the previous level. This allows players to progress at their own pace, thus reducing cognitive load as well as fostering motivation by building confidence and self-efficacy (Mayer, 2003; Schunk, 1991; Sweller, 1999). In addition, each level has what Malone (1981) refers to as “multiple level goals.” This means that at each level there are several goals or markers of success that players can try to achieve as they progress through that level. In Level 1, for example, where the overall goal is to navigate to the correct FMS pages during the preflight programming sequence, students are given automatic, and successively explicit, prompts by the virtual captain; there is also a countdown clock that records the amount of time they took to complete the sequence. For beginners, the goal might be to complete the programming sequence with minimal prompts; once they can do that, they can then still be challenged by trying to complete the task as quickly as possible. This way, as players master the steps of a particular level, they can still be challenged to improve their score and/or time. This type of multiple level goal system also means that more adept players, and/or those with more prior experience in the FMS subject matter, can progress though the initial levels quickly – yet still be challenged – while more beginning students can still experience a sense of accomplishment with minimal frustration.

As a player reaches the level of mastery required at one level – such as completing the task within a given time and with minimal errors – he or she then progresses to the next level. At this next level, the player is given more responsibilities, higher level tasks, increased workload, less help and/or faster play. Because the skills required to successfully program the FMS tend to build upon each other (it would be difficult to do error detection and problem solving if one did not know how to navigate through the FMS pages), each level must introduce new information when the player needs to know it. This way, the game is always kept at an optimal level of difficulty and complexity. For example, beginning players only need a few commands, but as they become more advanced they learn to use more complicated features of the system.

Another way to challenge players and keep them motivated is to introduce an element of **uncertainty** – not only about whether they can accomplish the designated goals in a given amount of time, but also about what might happen next. Having some degree of randomness, throwing in interruptions, unexpected events, and twists – particularly in the higher levels, when lower level skills have been mastered – not only serves to increase interest, but in the case of FMS programming, also replicates some of the demands that are placed upon actual pilots. As with many tasks, most pilots can perform FMS tasks adequately in routine, low-pressure situations, such as during an uneventful cruise phase of flight. But pilots can experience great difficulty and an increase in errors when they have to program the FMS in high-pressure situations, such as when readying the plane for take off, preparing for approach, or when some external event occurs, such as having to deal with poor weather or messages from Air Traffic Control (ATC) requesting deviations from the flight plan (Lee, Sanford, and Slattery, 1997). Our review of the ASRS incident reports indicated that FMS errors indeed occur often under these demands. In the FMS Programming Game, we’ve included an intercom that periodically buzzes, requiring that pilots and/or the virtual captain suspend what they are doing and attend to the external event (e.g., a flight attendant asking the pilot to deal with an angry passenger or ATC requesting the aircraft to divert to another airport). These added events increase the multi-tasking aspects of programming the FMS, where dealing with distractions is a requisite duty that pilots face all the time in operational conditions.

Finally, in order to increase the challenge, The FMS Programming game also incorporates **competition**. Not only will players compete with the clock to try to complete tasks with a given deadline or best their own time, they will also compete with other players. As the game is designed now, players will be able to access a scores page which shows the top 25 players’ scores by user name, best time at each level, and what level they have reached. This type of competition is in line with what we believe would appeal to our target audience for this game – pilot trainees. Based on our previous work with them, we know the student pilots in this program are competitive with one another, albeit in a friendly manner, so that public posting of scores would likely be a motivating element of the game.

Game Element 4: Feedback

- *Training Environment Characteristics:* The environment consists of several smaller tasks, supporting a larger task; some of the tasks involve procedural learning; some tasks involve communication with crew members.

⬩ *Associated Cognitive functions:* system perspective, task management, shared understanding, procedure execution.

⬩ *Recommended Game Element Variation:* Employ a variety of feedback methods, depending on the game level and the task type.

The FMS game utilizes several forms of **feedback,** most of which are relevant to FMS training. For example, players receive constant feedback about time, in the form of a countdown clock that is visible on the screen at all times. Time is particularly relevant in FMS programming. Thus, setting time limits and making time one of the criteria for moving to the next level are not just artificial constraints added to the game to motivate players; they are an important component of the real-world task and should be included in the game. Other forms of feedback include qualitative, descriptive or prescriptive comments from the virtual captain who responds to quick and accurate actions with comments like, “Excellent job! We should have this thing programmed in no time.” On the other hand, he will respond to slow or inaccurate actions with comments such as “That was pretty slow; maybe you need to go to the quick guide and study the steps a bit more”. Because much of the task involves procedural learning, feedback should be immediate at the lower levels, when the students are learning the correct procedures. Yet it is also important that students learn to develop error detection strategies. Therefore, at higher levels in the game, the explicit feedback should be more intermittent, allowing students to recognize, on their own, when they’ve made a mistake, and then giving them a chance to take steps to remediate it.

Some aspects of FMS programming can entail multiple steps, making it particularly difficult when they are first introduced for players to gauge how close they are to reaching the goal. Therefore, the lower levels of the game should include feedback in the form of a “steps remaining” checklist that is always visible on the side bar, and which enables players to see how many steps remain to be completed. Finally, at the end of every level players will receive a summary of their performance, including time and number and type of errors (if relevant). They will also receive a commendation memo from the virtual captain that reiterates what they learned in the session; this memo will either allow the player to progress to the next level or require them to repeat the level and/or complete a remedial task. Reminding players of the training objectives of the game and providing them with an opportunity to reflect on what they have learned serves to reinforce those objectives and reduces the likelihood that players will get sidetracked by the game itself and forget about the purpose of the training.

Game Element 5: Goals

⬩ *Training Environment Characteristics:* multiple component tasks supporting a larger task; moderate to steep learning curve; need to automatize skills and internalize procedural knowledge; varying levels of player skill proficiency.

⬩ *Associated Cognitive functions:* judgment, procedure execution, task management, timed response

⬩ *Recommended Game Element Variation:* Employ multiple goal levels and set criteria for level mastery

We have already briefly touched on the multiplicity of goals within each level, but the **goals** of the game can also change from level to level. At the beginning levels, the goals or objectives are generally procedural and well-defined (e.g., “Navigate to this page and enter this piece of data.”), but they become more conceptual and less explicit (“See if everything’s okay”) as the player’s proficiency increases. As the player reaches higher levels, the goals will generally involve detecting a problem, diagnosing it, developing a solution or several solutions, and then deciding upon and implementing the best one. Also, some levels may have “indirect goals” that are not as explicit, at least initially. For example, in the beginning levels, the player’s main goal is to navigate to the correct page. However, an indirect goal is to learn FMS terminology, particularly acronyms that describe certain items of information (such as BOW or Basic Operating Weight, and OAT or Outside Air Temperature). The players should acquire these terms as the virtual captain goes through the pages, but they won’t be explicitly asked about them until reaching the later levels when they are given the explicit goal of retrieving certain items of information. This type of goal structure should provide players with repeated exposure to information and procedures, without overloading their working memory.

The TARGET Tool – Preliminary Considerations

The process of linking game element variations to specific requirements of the training environment is what will underlie the TARGET Tool (**T**ool For **A**pplying **R**obust **G**aming **E**lements to **T**raining). As noted previously, one use of TARGET will be to generate testable hypotheses about which game elements are most appropriate for a particular type of training environment. Our inspiration for these hypotheses will be based from joint considerable of principles of cognitive psychology and on our findings from the serious game literature review, We will test and further refine these hypotheses in the series of empirical studies (described in chapter 8), which we plan to conduct in Phase II.

While formal development of TARGET as an XML entity will occur in Phase II, we devoted some Phase I resources to draw up possible models of how TARGET might be structured. Below is a brief overview of one possible approach to how the tool may work.

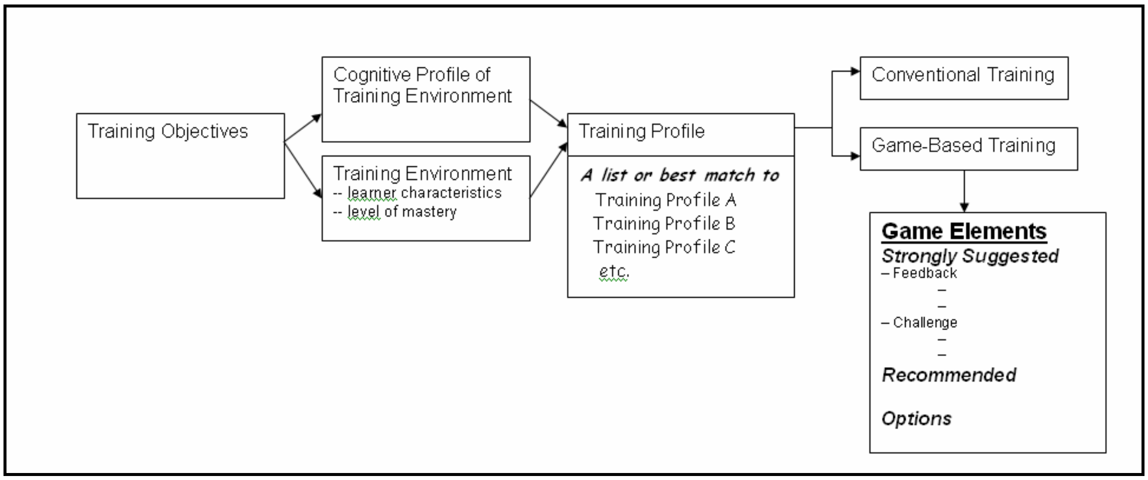


Figure 7. Possible version of Tool for Applying Robust Gaming Elements to Training" (TARGET).

We begin with the sketch in Figure 7. Our conception is that TARGET will be web-enabled and will provide instructional designers and instructors with guidance in (1) selecting particular elements of gaming to improve training, (2) identifying particular types of experimental manipulations to advance the literature, (3) collecting information from new empirical studies of gaming elements' impact on training performance, and (4) assisting designers in "tweaking" existing games for the purposes of training improvement. Because the tool can serve a number of purposes, we view it is as the best option for this solicitation.

In our preliminary vision of the tool, users of TARGET, primarily instructional designers, will be asked a series of questions designed to distill the key training objectives of the course in question. Based on their responses, the tool will generate a corresponding cognitive profile, listing the key cognitive functions that students need to engage in order to achieve successful performance. The tool will also generate a profile of the training environment (such as what resources are available, whether the course has pre-requisites, what problems are most frequently encountered, etc.). These two profiles will combine to form a "Training Profile". One option is for the tool to then select a "best fit" from a set of prototype training profiles. For example, "Training Profile A" may encompass training environments that primarily entail rapidly extracting, processing, accessing, and evaluating electronic data; "Training Profile B" may primarily entail selecting and effectively carrying out proper communication protocols; while "Training Profile C" may involve developing rapid perceptual motor skills, and "Training Profile D" may primarily entail developing strategies to manage task flow in a fast-paced multi-tasking environment. Which training profile the tool pulls up will depend upon the results of the initial training objective querying. Based on the Training Profile and other factors, the tool would then make recommendations about whether this area is a good candidate for game-based training, and, if so, which game elements have been associated with positive training outcomes in that situation.

As an example, consider the training profile description below, which would be used to characterize the FMS training environment. We would similarly construct an environment description for the procedure-callout

competency, with which another set of optimal game elements might be associated. Over time, we would expect our description of the different application environments to become more complete, with a greater variety of profiles defined, thus the tool's prescriptive ability would increase in scope and power.

The tool could also be used "in reverse" to help instructional designers evaluate whether a particular instructional game would be appropriate. In this scenario, users would also be queried about the characteristics of the game, then receive output about how well the game characteristics match the training profile. It was for this reason that we have considered an XML construction (with its inherent tagging capability) to be well-suited for this application.

As an aside, it should be noted that could employ other options for tool development in this project. The most obvious is to attempt to create a "generic" game whose elements could be systematically varied by means of programmed input. That is, this game would be computerized and would require subjects to master a set of (probably) timed responses under conditions of high stress. Variations in elements (e.g., feedback, competition, realism, goals) could be integrated into the game as a set of menu options that the user/designer would choose at the start. This game element test bed could thus serve as the vehicle for collecting empirical data in which game performance would be examined under different variations of levels of the various elements.

However, we have elected *not* to pursue this option for four reasons. One, it has no domain specificity, and thus may lack face validity with many potential user groups. Two, there is no assurance that any type of transfer across multiple settings will occur. Three, it does not contain any logical connection with the empirical literature on gaming element impact that has been collected thus far, nor does it permit any accrual of information as more studies are identified or conducted. Four, and most important, it does not address what we consider to be the core aspect of this solicitation—incorporating gaming aspects into an existing training curriculum as opposed to creating some *completely new* activity that students will have to spend time on in order to acquire the underlying core cognitive competencies that should ultimately transfer. As such, it becomes an added requirement, and hence a larger-than-desired footprint rather than an organic capability that can naturally integrated into the curriculum for a given domain application.

Training Environment A

This training environment involves operating a complex electronic device, including programming data, locating and verifying information, detecting errors and inconsistencies, and problem solving. The tasks are usually carried out in a high pressure situation with tight time constraints, multiple demands and interruptions, unpredictable events, and serious consequences for failure. Parts of the task are routine, and require automatized procedural knowledge; other aspects of the task are event-dependent and require factual and conceptual knowledge (both of the device itself and of the environment in which it is used) in order to generate solutions in response to various external events

Figure 8. TARGET profile description of the FMS training environment.

Chapter 5: Development and Specifications of the FMS Training Game

Based on our task analysis of the FMS competency and the subsequent crosswalk analysis, we developed a general structure for the FMS programming game as well as specifications for which game element variations were likely to be best suited for the FMS game. The next step was then to flesh out the specific training objectives that would be addressed in the game, and to develop content material for both the overall game play and the tasks and exercises that we planned to incorporate into the game. To do so, we reviewed the literature on FMS training issues to distill some of the more common difficulties that student pilots and practicing pilots experience – in particular, the errors they make while operating the FMS, and the typical

consequences of those errors. We also conducted more focused interviews with our SME's at ASU, again concentrating on the common difficulties students experience while learning to program the FMS, and the types of errors they make. We also worked with the SME's to generate several realistic scenarios and challenges that we could incorporate into the game. We then used this information to develop the design specifications and storyboards for the FMS game. We later ran the storyboards by our SMEs at ASU, as well as some SMEs at Mesa Airlines, who evaluated the features and utility of the game design in a formative evaluation.

Review of real-world FMS-related incidents

As noted above, in addition to the task analysis of FMS operations that we conducted in Task 1, we also gathered information from various sources about the typical problems that student pilots and professional line pilots experience when programming the FMS. The purpose of gathering this information was, in part, to help us flesh out our game design specifications and create realistic exercises We drew from a number of sources, including empirical studies on FMS training (Lee, Sanford, Slatterly, 1997; Salden, Paas, van der Pal, van Merrienboer, 2006), as well as incident reports on NASA's Aviation Safety Reporting System (ASRS) website (http://asrs.arc.nasa.gov/) and the Flight Deck Automation Issues database (www.flightdeckautomation.com). The information gathered from these sources demonstrated a real need for improved FMS training, even for experienced pilots. At first glance, FMS programming, or at least some parts of it, may seem similar to fairly routine data entry required of any computer system. In actuality, though, it is a vitally important task that affects airplane performance and navigation in a number of critical ways. Failure to correctly program, update, and verify information in the FMS can have serious consequences, such as running low on fuel, being in the wrong airspace, committing errors in navigation, and even crashing (e.g., through controlled flight into terrain—CFIT—accidents). The crash of American Airlines Flight 965 in Cali, Columbia, for example, was caused by pilots entering an incorrect waypoint into the FMS, taking the plane off course and into a mountain. In another incident, the tail of Singapore Airlines Flight SQ286 struck the runway, substantially damaging the plane and jeopardizing the lives of the 368 people aboard, after the pilot incorrectly input the takeoff gross weight as 247.4 tons instead of 347.4 tons.

Due, in large part, to the complexity of the system, and its non-intuitive interface, however, FMS programming is prone to many errors. Several of the reports we looked at emphasized the need for pilots to automatize their FMS programming skills as much as possible. A quick search of "FMS" on the ASRS website reveals several incidents that are a direct result of difficulties and errors in programming the FMS. This has also been borne out in experimental studies as well. In one study investigating pilot performance, eight out of nine significant route errors were due to errors in FMS programming, including incorrect FMS procedures being entered and the inability to enter the procedures in a timely manner (Lee, Sanford, and Slatterly, 1997). Some of the typical errors we distilled from the reports reflect: failure to perform inputs in timely manner, failure to recognize errors and inconsistencies, inputting information incorrectly or inputting the wrong information, and the inability to integrate information from within the FMS device. Distractions (such as flight attendants requesting the captain's attention, while he or she is the middle of programming the preflight sequence) also play a role.

This analysis of typical errors and difficulties not only illustrates the relevance of FMS training – and its potential application outside of ASU – it also highlights many of the errors we plan to address in our GBT product, and provides examples of real-world incidents and scenarios that we can include in our game design.

Focused Input from FMS Subject Matter Experts

We also met with our SME's at ASU and Mesa Airlines to obtain focused input on the main difficulties trainees have when learning to operate the FMS – difficulties that we could then target in our FMS training game. These SMEs – really super-experts – have years of experience in training student pilots how to operate the FMS. They not only offered useful technical information about the more subtle aspects of the FMS, they also provided invaluable insights into which skills students found most troublesome and which tasks are more difficult to perform. This information was combined with the results of our cognitive task analysis described in Chapter 3 to create a rich knowledgebase of FMS technical challenges, contextual conditions, and advanced operating techniques. We then used this knowledgebase to design the FMS game

environment – specifying tasks, challenges, instructional feedback, and scripts that would address those deficiencies.

Below is a sample result from these high-level SME interviews. This summary details some of the most common errors made by students who are learning to program the FMS:

Problems with the FMS Preflight Programming

Status Page:
1. One of the most common errors is simply forgetting to check this page.
2. If they catch it, and the page is out of date, many pilots don't know how (or even if it's legal to) update it.

Initializing the FMS:
1. Many times pilots will bring the current lat/long position to the set position instead of the airports lat/long.
2. Or they will bring the airport code (KPHX) down to the scratch pad instead of the lat/long for the airport.
3. Occasionally this has to be done twice for the FMS to reset its position.

Flight Plan:
1. One of the biggest errors is simply trying to build the route out of order.
2. The second is not putting the first fix cleared to in the "to" line if it's a vector departure.
3. If it's a SID (standard instrument departure) then they forget to put a transition at the end of the SID.
4. Many times they will put the flight plan in from the release paperwork instead of the actual clearance.
5. They will often forget to put the flight number into the FMS. (not really that important)
6. They will put the flight plan fixes "to" into the "via" line where the J routes and V routes belong.
7. Or the opposite, they will put the "routes" (J route or V routes) into the "to" side where the "to" belongs.
8. When they select a STAR (standard terminal arrival route), they again forget to add a transition fix.
9. Forgetting a transition fix will automatically create a "discontinuity" in the flight plan because the FMS does not know how to "tie" the arrival to the route.
10. As with any page, they forget to EXEC to save their inputs. So if they have to change something or delete something later, it will delete everything. (EXEC = SAVE on your computer at home).

Legs:
1. This is a simple check to make sure there is no discontinuity in the flight plan.
2. They should check and make sure the mileage is correct. This is a good indication that something was done wrong if it doesn't make sense. (6500 miles from Phoenix to Tucson)

Performance Initialization:
1. Most common error here is forgetting there are two pages that have to be set: PERF and PERF INIT.
2. The PERF page is pretty easy. Simply putting the ATIS temp in the boxes. But sometimes they use the actual outside temp instead of the ATIS temp which is required.
3. If they don't forget to go to the PERF INIT page, they forget to fill in ALL the boxes. Especially cruise alt. 4. Then they forget that there are two pages to this page. Next page is the winds and avg temp deviation. They get this info off the flight release.
5. They forget that twice a year the avg persons weight changes. Summer weights vs. winter weights.
6. They will use the actual fuel onboard instead of fuel onboard MINUS taxi fuel. Usually 200 - 600 lbs less.

Multifunctional Display Menu:
1. There really isn't a right or wrong answer for this page. It is kind of your own preference as to what is displayed on the MFD. Standard is Hi nav's, speed, and altitude. Then on the next page is the RNG to alt.
2. If there is an error possible it is setting the wrong side. (L or R). The student might be selecting their preferences onto the other pilot's side.

Radio:
1. The most common error here is trying to auto tune the FMS while the pilot is in "green needles". This is impossible to do.
2. The pilot has to be in "white needles" to auto tune the FMS.
3. Many times they will forget that changing the active Nav frequency will take them out of auto tune.

In General:
1. The FMS confuses many pilots simply because of the number of pages available. This is why it's important to activate each button so that the student could get "lost" in all the pages.
2. Often times they will try and do all this backwards or out of order, which usually leads to many errors.
3. Many students don't learn the "J-Pattern", which makes it difficult for the student to get a starting point and an ending point for that matter.
4. Many students don't understand that the FMS is a "flight management system", and that it affects many parts of the aircraft and the flight. (Navigation, auto pilot, performance, etc.)

A few of the most common errors that occur during flight:

Take off and Climb Phase:
1. Students forget to change the departure runway in the FMS if the runway changes on their taxi out.
2. Students forget that they have to "transfer the bleeds" to the engines in order to get the climb and cruise thrust settings.
3. Going "direct to" a fix, or "intercepting" an airway is always a challenge for new students.
4. They put a "direct to" fix into the "from" line select key instead of the "to" line select key.
5. Or.. they put a "from" fix for an intercept course into the "to" line select key.

Cruise Phase:
1. Forgetting to EXEC their inputs.
2. Arming Nav mode (auto pilot) to the FMS after they are cleared to a fix from a heading.
3. "grabbing" the wrong line select key. (they are pretty close together and the picture shifts in the airplane)
4. Not verifying their selections with the other pilot.

Descent Phase:
1. It is important that they make sure all the speed and altitude restrictions are in the FMS. (vertical Nav).
2. They forget to or select the wrong approach for their airport.

Approach Phase :
1. The MOST common flight errors happen in "sequencing" the FMS for an approach.
2. When they sequence the FMS for the approach they often forget to change the intercept course for the approach. We call this "making the numbers big", because the FMS gives you the correct course to fly on an approach and you simply have to push the crs intercept line select key. (very difficult to explain).
3. Making sure they have the right approach selected..

The SME-supplied knowledgebase described above also yielded a variety of ideas for creating realistic and challenging problem-solving scenarios. These could then put incorporated into the FMS game. Eleven of the most compelling problem scenarios for operating the FMS are summarized below:

Sample Problem Scenarios for FMS Operations

1. Normal Operations: Dispatch Release, ATIS, ATC Clearance – Program FMS.

2. ATC Clearance different than release. Use ATIS, ATC Clearance to program FMS; must decide if fuel sufficient for flight.

3. After programming FMS, possibly during taxi out to runway, there is a runway change. Must reprogram FMS with new ATIS, may generate a change to flight plan with different departure.

4. More passengers and bags than on Dispatch release. Your data dictates flying at a lower cruise altitude. Must program FMS and see if fuel is sufficient for flight.

5. Airborne – given short cut or reroute, must program FMS with new route or direct routing.

6. Weather at destination bad – must hold, requiring programming FMS with holding and decision about fuel and possible divert.

7. Divert – must reprogram FMS with new route to alternate airport.

8. Change of arrival at destination – must reprogram new arrival in FMS.

9. Loss of engine enroute. Must use FMS to calculate optimum cruise altitude and reprogram FMS for possible divert to new airport.

10. Weather along flight path dictates a new route – must program new route into FMS around thunderstorms, different altitude for turbulence or another route to avoid high headwinds from the jet stream. All these situations require looking at fuel to see if the flight can be completed safely with new FMS route.

11. Advanced use of FMS would be to add in abeam points during a direct routing. This allows more accurate calculations by the FMS when enroute winds are added to the abeam points that were eliminated with the direct routing.

Overview of the FMS Programming Game

Based on our the game element taxonomy, the cognitive task analysis of FMS operations, our analysis of the training needs at ASU, and our research and SME interviews on FMS training issues, we then designed the specifications for a serious game designed to target FMS training.

Below is a summary of some of the main features of the FMS programming game, which illustrate how the game elements and pedagogical principles we derived from the above work were implemented:

Purpose of the Game

The FMS Programming Game is designed to train players to program the CRJ-FMS quickly and accurately. The goal is for players to become so familiar with FMS features, page navigation, and procedures they are able to:

- program the FMS preflight quickly, with no mistakes,
- program the FMS during the climb and cruise phase of the flight, as new information comes in,
- program the FMS during the descent and approach phase, and
- detect errors and problem solve using the FMS.

Training Objectives

By the time the players reach the final level, they should be able to:

- navigate through the FMS pages quickly to find a particular page or data item
- understand the key features, terminology, and coding used in the FMS
- use the buttons and scratchpad to select and enter data quickly and without error
- identify what data and functionalities are on key FMS pages
- follow the appropriate sequence of steps when programming the FMS preflight
- make the proper calculations and inputs in response to changing situations
- carefully check and verify information as well as recognize and catch errors
- integrate the data from multiple pages in order to understand how making a change on one page affects data on another page, as well how it affects the plane’s performance

Structure of the Game:

The FMS game is divided into two parts: One is the main FMS Programming Game, and the other is an embedded game called the “Pilot’s Lounge” which players can access between levels of the main game.

The Main Game Summary

The "Main Game" provides whole-task training and takes place in the simulated cockpit of a CRJ Jet. It provides players with a context-rich, realistic environment where they assume the role of the first officer who must program the FMS preflight sequence, update the FMS as new information arises, detect errors and solve problems, all under the guidance of a "virtual captain" who provides prompts and feedback.

The Main Game Interface

The main display of the game is a view of the cockpit. Embedded in the cockpit panel are various instruments and tools that the player can enlarge and interact with, such as the FMS, the multifunction display (MFD), primary flight display (PFD), flight control panel (FCP), and other instruments related to FMS functioning, as well as a clipboard with forms and charts that players will need while programming the FMS. (See Figure 9 for a snapshot of the FMS Game interface.)

Figure 9. Snapshot of the FMS programming game interface.

On the left side of the picture in Figure 9, is the FMS device, which has behaves much like a real FMS, with clickable buttons and changing screens. Players can expand the FMS (as shown here) or reduce it by clicking on it. To right of the cockpit is a bar that contains several cues that inform players on their performance, as well as an intercom that gives players periodic contact with external entities outside the cockpit.

> *The Takeoff Status box* tells players how many minutes until their scheduled takeoff time. This time will progressively decrease as they ascend levels. The status box will also tell them how much fuel has burned while they were programming the FMS. At the end of the allotted time, they will receive a message indicating either that they are ON TIME (successfully completed the task in the allotted time) or they exceeded the allotted time and have to RETURN TO THE GATE.
>
> *The Intercom* is used to increase the difficulty of the programming tasks, mirroring the interruptions, distractions and pressures that pilots normally experience. Occasionally, the flight attendant or air traffic controller (ATC) will send the player audio messages asking them to perform some task, such as deal with an angry passenger or divert to another airport. When the intercom flashes, the players must press the button (immediately) to receive the message. Sometimes the message will require the player to stop programming the FMS and return to it later; other times the message will be new information that requires players to problem solve and reprogram the FMS.

The Status Board appears only in the lower levels. It tells players which steps in the preflight process they have completed – and thus, how close they are to their goal. It also serves as a mnemonic device.

The Taxi Way also only appears during the preflight programming levels. It shows a line of planes, including the player's, that are waiting to take off. The player's plane moves up in the line and closer to a yellow "take off line" every few seconds. This provides an additional visual indication of how close the player is to the take off time; it also adds stress to the player by highlighting how failure to program the FMS efficiently can impact the entire flight schedule.

At the top of the screen is the game control panel, with buttons that players can press to receive help or obtain additional information.

The *Task button* – opens up a window with text instructions for that level

The *Repeat Audio button* – repeats the most recent audio message/instructions

The *See Text button* – shows a text script of the audio for that page

The *Scores button* – shows scores for that player, as well top scores for other players

The *Help button* – provides links to information such as a review of the cockpit features, links to quick guides, and general FAQs

The Main Game Storyline

The player assumes the role of First Officer (the one primarily responsible for programming the FMS). Seated next to the player in the cockpit is a virtual captain. The captain interacts with the player in several ways, including audio, by pointing to certain features on the FMS, and, at the end of a session, by giving them written memos. (Note: Because the view of the cockpit is from behind and presents a somewhat first-person view, only the arms of the captain are visible.) The captain provides prompts and hints, local and global feedback (i.e., immediate and end-of-session summary), administers some of the positive and negative consequences, and models procedures and problem solving.

The Levels of the Main Game

The game is currently divided into seven main levels. As the player progresses through the levels, he or she takes on more responsibility and the captain provides less guidance.

Level 1: FMS Preflight Programming – Introduction

Overview: The player's primary task is to press the correct FMS buttons and menu links to navigate through the steps of the FMS preflight programming sequence quickly and accurately. The virtual captain provides a series successively explicit prompts, on a timed schedule (e.g., every 10 seconds). The first prompt just asks the player what step is next. The second prompt tells the player what step is next. The third prompt tells the player what page they need to navigate to. The fourth prompt tells them what buttons to press to get to that page. By the fifth prompt, the captain's finger points out the buttons. At this initial level, the players do not enter or program data, but are exposed to it by watching the virtual captain model the task while explaining what he or she is doing.

Training Goals: This level serves to familiarize players with the FMS display and how the function and line select buttons operate, as well as train them in the steps involved in preflight programming. By the end of this level, players should know how to use the function and line-select buttons, and should be able to navigate to the appropriate pages in the programming sequence with little or no guidance.

Game Goals: The player's goals are to remember the proper sequence of steps involved in preflight programming, what pages the steps are carried out on, and what buttons and links are needed to bring up those pages so they can press the correct buttons and navigate to the appropriate pages with as few prompts as possible – and before the allotted time runs out.

*Feedback***:** Players receive verbal feedback from the captain after every action. For example, if the player completes the first step (i.e., presses the correct button to get to the STATUS page) within 10 seconds, the captain says, “I’m impressed! You seem to have this step down.” If not, the player receives a more detailed prompt and another 10 seconds to complete the task. See Table 11 for an example of the feedback schedule.

Table 11. Example of Feedback for Level 1, First Step in the Preflight Process.

Time (seconds)	**Prompt (Captain Says):**	**Feedback (Captain Says):**
0-10	“Okay, what’ the first step?	Nice job! You remembered the first step. We’ll have this thing programmed in no time.
11-20	(pause) First, we have to check that the database is active.	Good, you remembered what page it’s on
21-30	(pause) That’s on the STATUS page	Okay, so maybe you didn’t know the step, but you did know how to get there
31-40	(pause) You can get there by pressing the line select key next to STATUS on the INDEX page menu	Good, but try to find it on your own next time
41-50	(pause + annoyed tone) Come on, you get to the STATUS page by hitting this line select key next to the STATUS option (finger points to correct buttons FMS panel).	Let’s pick up the pace or we’ll never get out of here.

“Time” starts when captain first says “Okay”

If the player gets to the page within the time range, the captain says the corresponding feedback line

If the player does not get to the page within the time range, he or she gets the next prompt

In Level 1, only the relevant buttons are functional on each step, so players cannot make errors; they are just scored for time. Players also receive feedback at the end of the session, where time either runs out or they successfully complete the task. This feedback takes the form of a letter of commendation from the captain to progress to the next level, or to repeat the level. He or she can also press the scores button to see a summary of the time it took them to complete the level.

Scoring: Players are scored on

- The number of prompts needed for each step in the programming sequence
- The total amount of time it took to complete the preflight programming
- Whether the player achieved an ON TIME take off (completed the programming before the plane reached the end of the taxi way) or had to RETURN TO THE GATE and repeat the level again.

Advancing to the next level: If the player achieves an ON TIME take off with minimal prompts, he or she earns a recommendation from the captain to move on to the next level. At this point, the player is offered the option of going on to Level 2 immediately, exiting and returning to Level 2 later, replaying Level 1 for a better time, or taking a break and going to the “Pilot’s Lounge” (described below) to decompress and play some mini games. If the player does not achieve the mastery standard for this level, he or she has to RETURN TO THE GATE to review the steps and begin the level again.

The other levels operate in a similar fashion, with the player taking on more responsibilities, completing higher level skills, handling more tools and information, and dealing with more interruptions and random events.

Level 2: FMS Preflight Programming – Rapid Navigation through the FMS

- Players navigate through the pages involved in preflight FMS programming at a more rapid pace, handing interruptions, and programming some of the data themselves. The virtual captain provides fewer prompts at a more rapid (i.e., "impatient") pace, and does not offer a detailed explanation of each of the pages. The captain still does most of the programming, but occasionally asks the player to do so. At this level, the players can also make errors (e.g., can press the wrong button or go to the wrong page).

Level 3: FMS Preflight Programming – Information Locating and Error Detection

- Players are required to rapidly locate and verify key information, compare and integrate information across various displays and pages, as well as perform error analysis and institute problem solving measures to remediate any problems. Typical questions from the captain include, "I just got an update on the OAT – it's 14, not 20. Can you go in and fix it?" or "I'm not sure we entered the SID (standard instrument departure) correctly. Can you verify it?" or "I think there may be something wrong with the waypoints, can you check on that?" or "Check that the LEGS page is okay." These all require players to apply the terminology that they learned in the earlier levels, to navigate to individual pages out of sequence, and to recognize when information is not correct.

Level 4: FMS Preflight Programming – Full programming (with the Data Table)

- Players perform all of the preflight FMS programming on their own, using the table of information pre-distilled from the dispatch release form. At this level, they will program two flights. During the first flight, the captain will point out errors and not let the player move on to the next step until the error is fixed. But during the second flight, the captain will let the errors "slide" until the player is completely finished with the preflight program. He will then just tell the player that there is an error, which must be corrected. (Note: depending on the type of error may let the plane take off and let the player see the consequence of the error).

Level 5: FMS Preflight Programming – Full programming (with the Dispatch Release)

- This is similar to Level 4 except players have to extract information from the dispatch release form rather than read from the table of data that was provided at the earlier levels. Players will receive initial instruction on where to locate the information within this rather unwieldy, confusing report.

Level 6: FMS In-flight Programming – Updating, Programming, Problem-Solving

- Players detect and respond to discontinuities and errors during the cruise phase of flight; update flight plans and other information in the FMS, and make other display changes in response to external demands (such as ATC orders or changing weather conditions)

Level 7: FMS Approach and Descent Programming – Updating, Programming, Problem-Solving

- Players detect and respond to discontinuities and errors during approach and descent; update flight plans and other information in the FMS, and make other display changes in response to external demands (such as ATC orders or changing weather conditions)

The Pilot's Lounge

Embedded within the FMS game is the "Pilot's Lounge," where players can go in between levels of the main game (Figure 10). It offers an opportunity for players to "decompress" from the more urgent pacing of the various tasks of the main game, while still remaining engaged in the game. The types of activities offered would still reinforce skills, but do so in a non-consequential environment and would mirror more traditional computer games, for example, having players maneuver a little plane and shoot down balloons or other object that are labeled with the various steps involved in the preflight process – in the correct order, of course. When entering the Pilot's Lounge, players can "talk" to experienced pilots and aviation experts to get more information, tips, and anecdotes about the FMS. The main feature of the Pilot's Lounge, however, is the Arcade, which contains six embedded "gamelets." Gamelets are short, fun, and intense games that reinforce (via part-task training) the harder-to-master component skills from the main FMS programming game. Figure 11 shows screenshots from three of the gamelets. The left panel shows the "Bomb Shooter" gamelet, which reinforces the sequence of steps involved in preflight programming. Here, bombs labeled with the steps of programming task fall from the sky, requiring players to shoot them down in a specific order.

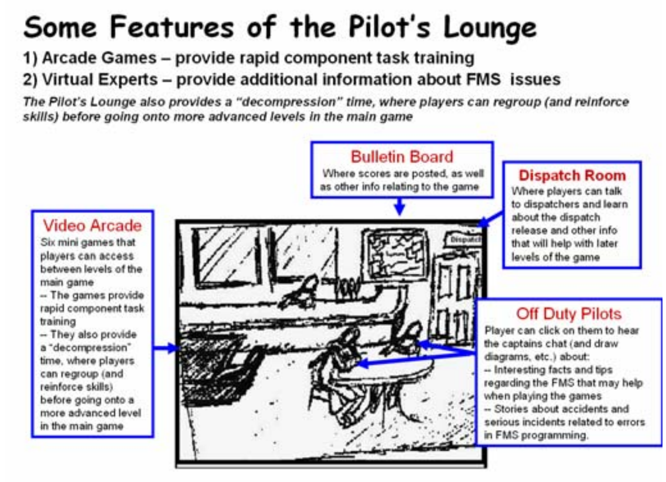
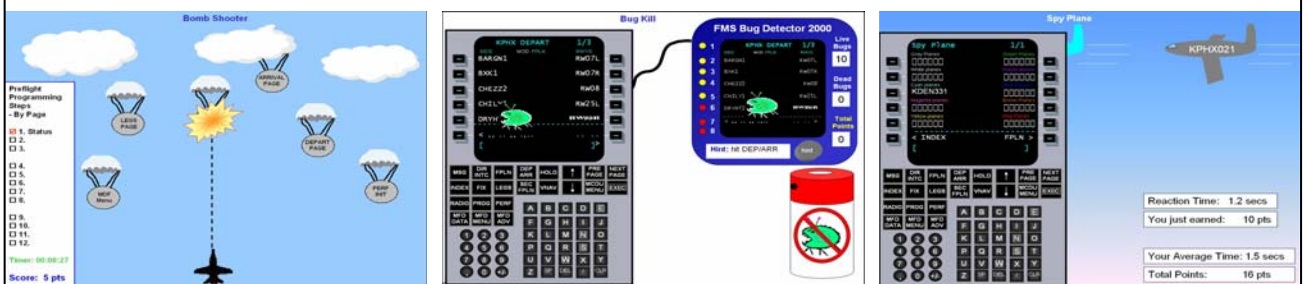


Figure 10. Features of the Pilot's Lounge.



(Bomb Shooter) (Bug Killer) (Spy Plane)

Figure 11. Screenshots of Pilot's Lounge part-task trainer arcade games.

In the middle panel is the "Bug Kill" gamelet, where players must use their knowledge of FMS organization to navigate to pages and "kill a bug" before it eats all the data on that page. The right panel is the "Spy Plane" gamelet which requires players to use the FMS keypad to rapidly type in codes painted on the sides of circling spy planes. Space constraints preclude our detailing specific design patterns here, but it is clear that each gamelet contains a number of distinctive design patterns that could be applied to other domains. For example, some of the display features could be swapped out to address domains that share common elements with FMS operation. Thus, we could substitute torpedoes for bombs, an inventory control device for the FMS, and a conveyor belt conveying parts for the circling spy planes. Using this replacement logic would allow the gamelets to be readily modified and adapted to a host of similar tactical problem solving settings.

The FMS Game Storyboards and Mini-Demo

The design specifications were then used to create a storyboard of the proposed game. The storyboard is a concrete representation of the game whose feasibility our SMEs can evaluate, and which our staff programmer can use later in the transition period and in Phase II to create a fully functional prototype game. Currently, the storyboards are presented on PowerPoint slides that illustrate what the game will look like, what functionalities will be present, how the game will flow, and what the players will experience. (The storyboards can be viewed on our company's ftp site at ftp://anacapatechnical.com/. The username: **asigbt** and the password is **asigbt06**.)

Note that due to file size and functionality constraints of PowerPoint, while the slides do have some interactive hyperlinks that allow simulated FMS button pressing, they do not illustrate all of the features we plan to include in the actual web-based game, such audio of the voice of the virtual captain or the sound of

the engines (though the script for the audio is shown in textboxes at the bottom of the screens). Also, some of the cues – such as the timer and the taxiway – are not functional in this PowerPoint version. Therefore In order to give our evaluators a better feel for how the game play might flow, we created a short demo using html-flash in order to illustrate some of the animated features of the game – things that might not be fully conveyed in the storyboards. It is not a prototype of the game, but rather a brief demo of what a moderately skilled player might experience during level 1 of the main game. A link to the demo can be found at: http://www.anacapatechnical.com/projects/fms/

The storyboards and mini demo were presented to a panel of SMEs – primarily instructors at ASU and Mesa Airlines, in order to get their feedback on the content, design and value of the game. This feedback was captured informally, via a group discussion, and more formally using a formative evaluation process, which is described below.

The Formative Evaluation Process

Formative evaluation is a method of judging the worth of an intervention or product whose development activities are still being formed or are on-going (Allen, 2003). It is the logical precursor to a summative evaluation, where the latter entails administration of the complete treatment to a representative group of users and assessment data are collected (Scriven, 1991). While formative evaluation is often viewed as being somewhat less formal than its summative evaluation, this is not necessarily the case. Indeed, one can collect formative evaluation data using structured materials, and then analyze and distill that information using systematic means. Such a process was followed in the present project in the following way.

A four-page survey instrument was designed to structure the review of our FMS programming game storyboards. The instrument, which appears in Appendix B, was completed by five participants who were employees of Mesa Air or ASU-Mesa instructors. All were highly experienced airline pilots who had extensive background in student training and were very familiar with FMS operations in the CRJ aircraft. Each participant reviewed the PowerPoint storyboards in some detail before completing the survey.

The survey covered such topics as technical accuracy, functionality, features, potential to be fun, and applicability to other tasks, among others. Five-point behaviorally anchored rating scales were used for each item. A separate question had users rate the importance (again on a five-point scale) of a number of features, such as interactivity, time pressure, challenge, and feedback. Finally, several open-ended questions addressed positive and negative aspects of the FMS game storyboard, including realism and the believability of the virtual captain.

FMS Game Features and Functions. The first five questions of the survey addressed the *content* of the game and whether it accurately reflects the technical aspects of FMS operation. In this regard, all participants rated the FMS game as being either “very high” or “high” in terms of its technical accuracy, its ability to give the user a good feel for the FMS, and its ability to cover important FMS functionality. Typical comments were that the game was “realistic,” “the pages that come up work the way they are supposed to,” and that “most of the difficulties of the FMS are covered.” Regarding the game’s potential to facilitate students’ FMS training, three of the participants rated this aspect as “very high” and the other two as “high.” As for the game’s fit with the ASU-AMT training philosophy, four participants rated this as “very good” and one as a “good” fit. Representative comments here are that the game “will increase he students’ desire to learn,” “it will really help familiarize students with the FMS,” and “this will help” ASU-AMT’s emphasis on FMS training.

FMS Game Characteristics. Questions 6-10 of the instrument addressed the characteristics of the FMS game from the standpoint as a game. Regarding the game’s potential to maintain students’ interest level, three participants rated it as “very high” while one each rated the game as “high” or “reasonable.” In terms of potential to be fun and engaging, three participants rated the game as “very high” and two as “high.” The typical comment here is that the game should be “fun and engaging” and that the Pilot’s Lounge Arcade should be “entertaining and educational.” When asked if the game has the features needed to be a successful game, three participants did not have enough information to answer the question, one participant responded “absolutely” while the fifth indicated “perhaps.” At this early point in the process, there is frankly more actual development of the GBT infrastructure needed before we can reasonably expect participants to formulate an informed opinion, though the potential would seem to be high.

On the other hand, when asked if they would recommend the game to other student-pilots, four participants responded “absolutely yes” with the other one indicating “yes.” Three participants responded “absolutely yes” in terms of using a similar gaming approach to our second competency, profiles and callouts, with the other two participants responding “yes.” Representative comments include “we need this at ASU now,” “any student should benefit” from this game, and “there is great potential to apply this to the other skills we are teaching.”

FMS Game Feature Ratings. We then asked our participants to rate the importance of nine features for being included in the FMS game. Interestingly, eight of the features were rated as “very important” or “important” by all five participants. These features included interactivity, immediate feedback, challenge, time pressure, command sounding voice-overs, realistic graphics, animation, and sound. For only one feature, publicized scores, was there equivocation, as all five participants responded with “perhaps.” Clearly, then, we will need to explore the desirability of posting everyone’s FMS game score in a public forum. Alternatively, it might be acceptable to have the high score (or perhaps several high scores) and player posted, with an update given (say) every week. No other scores would be listed. In fact, in terms of the notion of peer competition, the most frequent response was “perhaps,” suggesting that a closer look at the pros and cons of using this strategy as a motivational tool is warranted.

Other FMS Game Questions. The remaining four questions on the formative evaluation survey were open-ended. Regarding the features that participants liked best, typical responses included “realistic graphics,” “voice-overs,” and the “concept of an arcade game to teach part-tasks.” On the negative side (regarding confusing or distracting features), participants responded with caveats that “navigation through the different pages must be clear and consistent” and that the improvised clip board should be changed to “ship papers that would include ATIS, release, and ATC clearance.” This latter feature will be addressed specifically in later versions of the game.

When asked about their reaction to the realism of the virtual characteristics (the captain, ATC), most participants thought it was either “realistic” or “realistic enough.” However, there is recognition that the level of believability can be improved over time. The voice-overs used in the prototype are clearly just place-holders for more professional- (and pilot-) sounding ones that will be used in later versions of the game.

When asked for other, general comments about the game, participants responded that the game has “lots of potential” and is “going great so far.” All feel that a GBT concept for FMS will fill a present “training gap” and will be a good addition (“we need this now”) to the mix of training technologies presently being used at ASU-AMT.

Conclusions. In summary, the formative evaluation data strongly supports the conclusion that the FMS programming game has considerable potential, will be well-received at the department, and has features and functions that will provide a high-quality, believable game-based trainer for student-pilots. Ratings of the game’s technical accuracy, realism, and game-like potential were quite high for nine of the questions (discounting question 8, does it have the features to be a successful game, which 3/5 participants did not respond to), with 96% of the responses (43/45) occupying either the highest or next to highest rating category. From a statistical standpoint, the likelihood of receiving ratings this skewed from an underlying rectangular distribution (i.e., all five rating points equally likely at .20) is less than .001 (chi square = 68.04, df = 4), which is statistically significant. Thus, we are very encouraged by these initial data and believe that the development and implementation of GBT technology at ASU-AMT will be well-received and will fill a much-needed void in their training program. As described elsewhere, we are collecting informal confirmation that a similar reaction will be obtained in related technical domains, both in the Navy and elsewhere.

Summative Evaluation Framework

The summative evaluation to be performed in Phase II will be conducted within the logical framework established by Kirkpatrick (1959, 1960). Kirkpatrick’s four-level model has been the military’s primary approach for assessing the effectiveness of training interventions—be they a revised program of instruction, improved courseware, or a new training device—for the past 40 years. Within this framework, an intervention’s impact on the total training environment must be decomposed into the trainee’s perceived value of the training (Level 1), the degree to which the to-be-trained knowledge, skills, and attitudes (KSAs) are actually learned (Level II), the availability of the targeted KSAs for use on the job (Level III), and the

extent to which the organization's ability to achieve its desired objectives (e.g., increased flight safety) has been facilitated (Level IV).

The methods used to evaluate the summative impact of a training intervention are quite different across the levels (Kirkpatrick, 1996). Typically, some type of student-user critique serves as the data source in Level 1 whereas a pre/posttest comparison of KSA acquisition is the modal technique for Level II (Nullmeyer & Spiker, 2003). From a research standpoint, Level III is the defining feature of effectiveness, as demonstrated by a positive transfer of training to the job (flying) environment. These assessments will require some type of measurement procedure that spans both the training setting and the eventual operational (job) environment. This can either be done through observation of job performance, retrospective accounts by gaining-unit personnel, or some combination of both. Level IV assessments will judiciously sample an organization's performance at times prior to and subsequent to the intervention's incorporation into the daily running of the organization, where some type of cost effectiveness analysis and/or return on investment (ROI) may be calculated. Given the added complexity (more resources, longer time) of the evaluation at the higher the levels, it is not surprising that the vast majority of reported summative evaluations are conducted at Levels I and II (Bell & Waag, 1998).

Though the Kirkpatrick model has been uniformly embraced within the services, it is equally applicable to the commercial aviation environment. Indeed, this environment is highly similar to the military in many important respects, including having: (1) definable and measurable KSAs, (2) a discrete student population with motivation to acquire those KSAs, (3) a clear-cut operational environment (commercial flying) once training is completed, and (4) a specific organization (the particular airlines the students go to work for) that receives the trained students as new employees. For these reasons, we will be describing the conceptual, methodological, and empirical details of our Phase II evaluation within a Kirkpatrick framework. This will be described in the full-up Phase II proposal.

Chapter 6: Initial Specifications for Profile-Based Callout Skill Development Game

We elected to expend the bulk of our project resources on developing a storyboard and demonstration prototype for the most pressing competency, FMS programming. Nevertheless, we believe that significant training benefits would accrue from pursuing a staggered development strategy with the PBC competency, wherein high-payoff GBT concepts established with the FMS could be transferred to PBC training. With this strategy in mind, we want to lay the foundations for our second competency by formulating some initial specifications for how its training could be "gamed up."

To that end, the training would be organized around the four tasks which themselves are arrayed in order of difficulty. The first two tasks entail individual CM training. The third task would be trained with a "simulated" CM while the fourth task would entail actual crew training with a second, live, CM. In each case, the gaming aspects of the training will benefit from the TARGET taxonomy discussed earlier. The taxonomy would be used to select and define the high-payoff gaming attributes to heighten motivation, increase time on task, and enrich learning. We would use a similar interface, with the game icons surrounding the training display on an upper panel and outer right portion.

Task 1: Detect Profile Situation. The key to skill development on this first PBC task is to be able to quickly detect when a profile situation has been or is about to be encountered. A three-level progression of skill acquisition would form this game, where users would acquire associations with three different depictions of a profile: (1) the graphic view of a profile as illustrated in Figures 12-13, (2) an "outside" view of an aircraft maneuvering consistent with one of the designated profiles, and (3) an "inside" view from the cockpit instrumentation when a profile situation is encountered. Each level is briefly described in turn.

Level 1: Speeded response test of "profile" recognition. The beginning level of the game would entail showing a series of graphics that would display side-views of aircrafts that would either be in a profile situation or not. These would appear like that in Figures 12-13, but would be unlabeled. The objective is to train students to rapidly respond yes/no (using, say, the up arrow key for yes and down arrow key for no. Half of the graphics would depict profile situations (normal takeoff, landing, single engine approach, emergency descent) and half would not (e.g., cruise, turning past waypoint). A timer would start once the graphic comes on, and the student would earn points by (1) responding quickly and (2) accurately. They

would lose points for being incorrect or too slow. The speed criterion would shift within the level, so that with repeated "trials" the student would have to respond faster to gain points and keep from losing points. Graphics would have to be created for the non-profile situations, and would be constructed similarly to the profile situations so that students could not be using incidental cues to make their response.

The game interface would draw from the logic and layout of the FMS programming game, and as such, it will benefit greatly from our Design Pattern methodology to judiciously re-use high-payoff design concepts. This aspect of our company business plan – using our Design Pattern model to apply game concepts to new use cases and content domains – is described elsewhere in this report. Figure 14 depicts a schematic of how the interface for this level, and indeed the other levels and tasks, would be laid out. The student would see a running tally of points in the upper part of the display along with elapsed time. The game itself would be displayed below and to the left. An airplane icon is shown in the right side bar, which would move up on each trial where the student meets the accuracy and speed criterion, and it would move down when that criterion is not met. Once the student has succeeded in "passing" enough trials in a row, the airplane would be at the top of the display and the student would see the airplane "fly into" the gaming area and zoom off the screen. Sound feedback would occur after each response, with a pleasant sound occurring when the student has met the response criterion (speed, accuracy) and an unpleasant sound (e.g., airplane tires screeching on the runway) when the criterion has not been met.

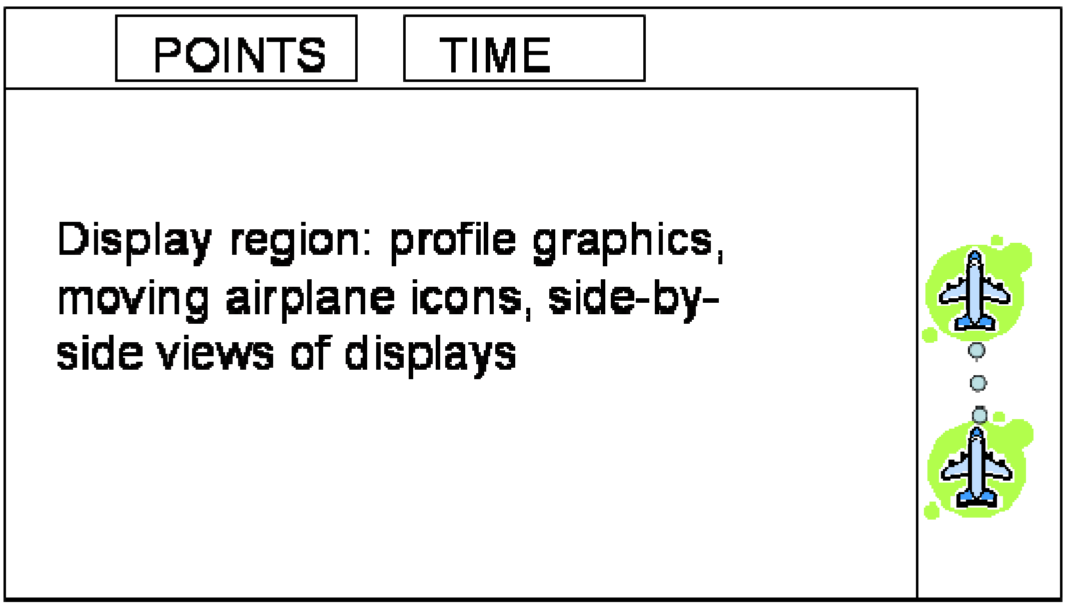


Figure 12. Schematic depiction of the profile-based callout game display.

Level 2: Speeded response test of "profiles" occurrence. Once students have completed the trials in Level 1, they will be moved to the next level. For this level, students will see animations of aircraft moving (using Flash) across the gaming region of the screen. The aircraft's maneuver will be the cue as to whether it is in a profile situation. As before, students are to respond yes/no (up or down arrow key) if they think the aircraft is in one of the 16 CRJ profile situations. The gaming aspects will be as before, with criterion levels of speed and accuracy required to make the airplane "move" in the right part of the display. The point once again is to force students to respond rapidly in order to gain points, as a way to promote overlearning and automaticity. When the student has reached the final criterion, the airplane will have moved all the way to the top of the right hand part of the display and will fly into the gaming area. Sound feedback would be similar to that used in Level 1.

Level 3: Speeded identification of profiles from aircraft displays. This third level is considerably more challenging, as it requires students to identify which profile an aircraft is in based on the information being displayed on the aircraft's instrument panels. This corresponds to the "inside" view discussed previously. Each trial would start with a question displayed on the screen, such as: Is the airplane in a non-precision approach (one of the 16 profiles)? Then several aircraft instruments would come on the screen, such as the

primary flight display (PFD) and multifunction display (MFD), with information that might or might not correspond to the queried profile. Again, students would respond rapidly yes/no using the up/down arrow keys. Criterion levels of accuracy and speed would again be the determinants of whether students "pass" that trial, which in turn moves the airplane icon (on the right part of the display) up or down. The stimuli would be a random mix of the various profiles and in some cases would not correspond to any profile. Feedback and motivating aspects of the display would be similar to those of the other two levels. Figure 15 shows an example of the type of stimuli that might be used in this level.

Figure 13. Depiction of the Level 3 display for profile-based callout game.

Task 2: Initiate Flow. Game-based training on this task is designed to instill overlearning in students about the cues that initiate each profile. While the PBC game will not train the pilot's flow tasks per se, it will attempt to instill knowledge about the sequence of flow tasks for each CM as well as identify points in the profile where a flow-callout sequence should be started. The three levels of this game address skill development in knowing (1) what starts the callout for each profile, (2) when in each profile does the callout sequence begin, and (3) the sequence of flow tasks for each CM in each profile.

Level 1: Identifying triggering events. For this and the other levels, the game will again reward the student with points for answering as correctly and as accurately as possible. The surrounding "game shell" will be as in Task 1, with times, scores, and the upward/downward moving aircraft icon present to inform and reinforce the player. For all three levels in this task, the game display will start by informing the player that "You are now in Profile X" (e.g., Missed Approach). In Level 1, immediately thereafter the screen will display the question: "What starts the callout?" The screen will then display a four-foil multiple choice item where the player's task is to pick the correct triggering event. For the example of Missed Approach, the choices might be:

a. landing gear up

b. reach decision height (correct)

c. flaps up

d. speed bugs set

After the player responds (a-d, using keyboard), they receive accuracy and response time feedback. If both criteria are met, they receive a distinctive message and pleasant sound, along with the aircraft icon moving upward. Failing one or both criteria has the opposite effect, with a negative sound (e.g., tire screeching) resulting. A series of these questions will be presented, with the Profile indicated first and the “what starts the callout” question asked next.

All profiles will be covered in these questions, with each profile presented several times (but with different distractors). Interspersed with these questions will be a second type of item that displays a task flow sequence and asks the player to identify the responsible CM (i.e., pilot flying - PF, or pilot not flying - PNF). To illustrate, if we were still in the Missed Approach profile and displayed the sequence: Gear, Flaps, Anti-Ice, the player should respond with PNF (say, by pressing the down arrow key). Again, the response time criterion would become more stringent over trials, requiring the player to respond even faster in order to move the airplane icon up the screen.

Level 2: Identifying action sequence starting times. Level 2 would also begin with the player seeing a statement like: “You are now in Profile X (e.g., ILS Approach).” They would then see a graphic representation of an action timeline and would be asked to pick the best time or upcoming action to start the task flow sequence. Again, an accuracy and response time criteria would be established that determines whether the player is “successful” on that trial. An example of the timeline graphic is shown in Figure 14. By clicking in one of the segments, the player would be making his/her choice (correct option here is “nearing the IAF.” A series of these action-timeline graphics associated with the various profiles would be presented in a random order. The student would be required to sustain an accurate/fast response pace to keep the aircraft icon moving upward and eventually passing out of the level. There would be multiple repetitions of the profiles over trials, where students will be expected to enter an overlearned state before advancing to the next level.

glideslope capture	gear down	FAF	procedure turn	flaps set	IAF

Figure 14. Schematic of an action-timeline sequence for missed approach.

Level 3: Recognizing flow task sequences. The hardest level for this task will ask students identify the correct sequence of flow tasks for each profile. The sequences will be displayed in a checklist type format, with either the PF or PNF’s flow designated. Accuracy and response time criteria will again be applied to determine success, with visual and auditory feedback the same as the other levels. The intent here is to instill in students an automatic recognition of the task action sequences that make up each profile from the perspective of both CMs. A common format would be used through the level, where the trial would start with: “You are the PF (or PNF).” Which profile is associated with the following task sequence? They would then see a display like that shown in Figure 15. The left part of the display shows the task sequence whereas the right part shows a table of the sixteen CRJ profiles. The student responds by clicking the appropriate cell as quickly as possible. Over trials, the response time criterion would become more stringent in order to achieve success, and once again, players would be required to sustain a fairly rapid, accurate pace of responding to advance out of this level.

Task 3: Single Crew Member Callout Training. The PBC game for Task 3 is played by students individually. The objective of this game is to instill proficiency and fluency in verbalizing the callouts associated with each profile. Each trial will start with the screen displaying the name of the profile and the crew position they will be assuming, such as Rejected Takeoff – Captain. Then, a scripted animation will appear in which the profile graphic “unfolds” over time. Referring to Figure 13 as an example, the trial would start with the airplane icon on the ground. The graphic would then reveal the first two actions the pilot performs: TOGA and stand up thrust levers. Then a blank would appear, and the student would say “set thrust.” The PNF’s task actions and callout would appear, in sequence. Then, the screen would display: PNF – “80 knots,” to which the student would verbally respond, “checked.” In like fashion, the rest of the profile would unfold over the display, where every place the PF is expected to say something the animation would pause, until the student makes the appropriate verbalizations.

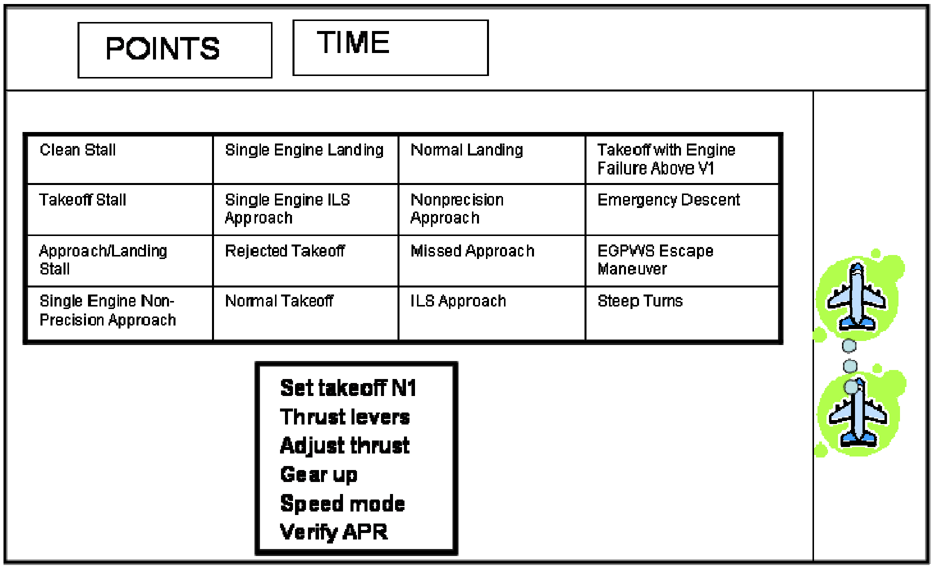


Figure 15. Schematic of a flow-task sequence identification trial.

Implementation issues. This game could be programmed in several different ways, depending on cost and implementation preferences. On the one hand, a (speaker-dependent) voice recognition capability could be employed that would be pre-set to recognize the precise callouts at the appropriate points in the profile. The system would stay frozen until the student utters the callout precisely as scripted. The system would record both response and number of attempts. Criterion levels of performance (time, accuracy) would be pre-defined and would dictate the student's success/failure outcome on each trial. Alternatively, the student could simply type in the callout using the computer keyboard, with all other aspects of the gaming environment the same. This second method is less preferred since it is not the actual medium (typing vs. voice) of the response in the transfer environment. In addition, to improve the fidelity of the training we could implement a synthetic voice system (or tape-recorded messages of actual pilots) that would speak the callouts of the other CM. This would likely improve transfer of training since the student would be learning to associate his/her own callouts with the voice of the other CM, which is more representative of what actually happens in the cockpit.

Levels of difficulty. The difficulty levels of this game would correspond to the pace of the clock to which the unfolding profile is synched. To begin, the pace would be slow, where events would occur much slower than they would in real time. These "below real time training" rates would start with something like 1/10th actual speed and would gradually speed up as the student masters a given level by sustaining criterion level performance. Eventually, the student would reach a 1-1 real-time level of stimulus presentation. However, once that pace is mastered, we would continue presenting the profiles (in random order) at even faster paces, to what is called above-real time training (ARTT). Use of ARTT procedures has proven effective in teaching tactical skills to fighter pilots (Crane & Guckenberger, 2000), and would show promise here as well. The game would continue until the student is able to maintain criterion performance at an ARTT pace of 2-3 times faster than real time. If the student is able to perform consistently at that level, then the callout verbalizations have clearly been overlearned and should transfer positively to the operational environment.

Response measure issues. To improve the impact of the experience, it would be advisable to "plus-up" the measurement package by including observer ratings of the student's callout verbalizations. Because a pilot's manner of speaking is strongly linked to their professional stature, it is important that pilots be trained to talk on the radio and to each other in a crisp, clear style. To that end, as students engage in the game, we should consider having an experienced pilot serve as an objective observer and evaluator of their performance. We could have the observer rate the callouts on each trial on a number of criteria, such as cadence, clarity, and timing, among others. Criterion levels would be established for these rating items as

well, and would be factored into deriving a “passing” score on each trial. Thus, the ratings would be combined with the time and accuracy measures to create a composite score that would determine the pass/fail outcome on the trial.

Task 4: Dual Crew Member Callout Training. Unlike the previous three tasks, this final task will entail true dual-CM training and hence will be a multi-person game. As in Task 3, the setup entails picking a profile with each student given either the Captain or FO crew positions. They would be given a background scenario, such as “you’re flying around a thunderstorm while approximately 125 miles due south of Wichita Kansas.” The instrumentation would then be set up to begin a particular profile (e.g., Steep Turn). The two students would then go through the entire profile, performing their individual flow tasks in the proper sequence and making their verbal callouts using the correct terms and at the right times. A combined time-accuracy criterion would be applied to determine “success” on a given trial. Unlike the previous tasks, where the profiles were given to players in a random order, a given profile in Task 4 will be performed repeatedly until some stable criterion of consistent performance is met. This is because the requirement to have both players perform at a sufficiently high level is more demanding, and will undoubtedly take repetitions until the criterion is satisfied. Once a profile has met the criterion, the players will be given another profile to perform and the process will be repeated.

Implementation issues. There are several ways this game could be implemented. The most straightforward way would be to use ASU-Mesa’s existing King Air simulator as the platform. This device is a medium-fidelity fixed-based simulator that has a three-channel visual scene database along with functional replicas of the primary flight display, radio, and multi-function display. Both crew positions are represented, so it would be a very effective way for students to practice their callouts and perform their task flows in the context of the different profiles. To “game” it, an instructor would need to be present to control the pace of events and to provide the criterion scoring to determine success or failure on each trial. The points earned would need to be tallied offline and provided as feedback to the students after each profile run. Difficulty levels could be instilled by forcing the students to make their callouts faster and more stylized over trials. However, it would not be possible to do any ARTT since the simulator is not step up for having information change on the instrument displays in other than real-time.

Alternatively, a two-person game could be programmed into a workstation where players would see a profile unfold on the screen much like in Task 3. While this would have lower visual fidelity and would thus not be as good for practicing their task flows, the system could systematically vary the pace of events, allowing players to start out with events occurring at below real-time speeds, then gradually increasing in rate until actually reaching levels above real-time. With a programmed game environment, we would be able to display times and scores on the screen, and supply written and auditory feedback after each trial. A combined time and accuracy criterion measure would be calculated to determine pass/fail status on each trial.

Response measure issues. Because the PBC task entails interdependent behaviors between two CMs, it will require observational scoring to capture the full complexity of players’ actions. Hence, we would have an experienced pilot (probably an instructor) serve as a the rater during each trial, regardless of whether the profile stimuli are implemented in the King Air simulator or programmed via workstation. On each trial, the observer would rate (using a five-point scale) each student on timeliness of the callouts, intonation, clarity, flow sequence, correctness of terminology, and cadence. A measure of crew coordination – how well the two worked together to execute the profile – would be collected as well. A points system would be established where ratings, total response time, and accuracy (e.g., false starts, hesitations) are combined to yield a criterion index that determines pass/fail for that trial. If the King Air simulator is used, there is the added ability to videotape the crew which could then be played back to get a more complete performance assessment.

Special challenge. With a little ingenuity, we have the possibility of adding some additional game-like characteristics to the PBC training to further enhance the students’ learning experience. Specifically, if we used the King Air simulator as the implementation platform, we could hold a “Callout Challenge.” This special event would occur after students have had an opportunity to go through all the tasks described above and pass the criterion levels of performance. At the Challenge, pairs of students could compete with another to see who can achieve the highest level of performance, which would result in some type of prize

(e.g., the Department could offer a cash prize or some other publicly endorsed inducement). As each pair performs a designated profile in the simulator, a panel of judges would render a verdict just like in the American Idol TV show. Also, the other students could be sitting in the back of the room and could also vote like the American Idol audience does. The judges would look for smoothness in the contestants' cadence, accuracy, professionalism, and teamwork, among other things. There could be both a winning crew and a winning individual. The winners would receive departmental recognition and would appear in ASU publications. There are also places on the walls of the department where photographic memorabilia are displayed.

Chapter 7: Applying Serious Games to Navy Training Environments

During the course of our Phase I research, we were mindful of the fact that our ASU-Mesa testbed is not a Navy training environment, though the two shares many similarities, particularly in terms of the types of technical training offered and the level of professionalism required of the trainees. In light of this, we made a deliberate effort, while carrying out our research, to insure that that serious game principles and elements we were creating could be successfully migrated to Navy training environments. Specifically, we identified analogous Navy courses that required similar competencies as that of FMS operations and Profile-Based Call-outs. We also created the game specifications using a design pattern methodology in order to facilitate the application of serious game elements to other domains. Finally, we identified Navy claimants who were ready to receive and use this application once it is developed in Phase II. Each of these three factors is described in more detail below:

Identification of Navy Analog Courses. Our decision to work with ASU-Mesa during our Phase I research and to used them as a testbed in our Phase II work was based on two important factors: (1) working with ASU provided us with the opportunity to directly test new training concepts on actual students who were enrolled in an actual training program, and (2) the competencies we will be training can be applied to a multitude of Navy training courses Indeed, this was one of the criteria used to select our two competencies of interest. To that end, we have reviewed the list of 4682 e-learning courses currently available from the Navy (supplied by Curtis Conkey, 14 Sept 06) for the purpose of identifying individual courses where component skills from one or both of our target competencies will be applicable. While this is admittedly only a cursory assessment of potential applicability—detail on specific training objectives would be needed for a more formal determination—it at least offers a first look at how our empirical data would be useful in informing Navy training providers.

Improved training in FMS operation will be designed to facilitate system mode identification, troubleshooting and diagnosis, adoption of a systems perspective, development of a "mental model" of system operation, and error correction, among other skills. As such, development of gaming elements for this competency should be applicable to any course where students must learn to operate equipment or a system for which a computer is a central logical component. From the list of e-learning courses, we identified 75 courses for which this might apply. These include such classes as basic radar operation (NIDA-9395TD02), digital test equipment (NIDA-93954104), applied digital electronics (15739), oscillator operation (NIDA-93953502), multi-meters (NIDA93951202), and SATCOM operation (NPGS-SATCOM12-1), among others.

Training in the callout/flow competency will be based on GBT elements designed to promote communication (including assertiveness and listening), crew coordination, and teamwork, and. We identified 67 e-learning courses where one or more of these component skills are likely involved. These include such classes as communication skills (COMM0005/6), team conflict resolution (TEAM0214), effective team communication (TEAM0172), and interpersonal communication: listening skills (45003), to name a few.

Based on this admittedly preliminary review, it is apparent that jointly focusing on a systems operation competency and a team-communication competency will give us considerable generalizability in the empirical findings we collect. During the transition task, we will analyze the Navy training curricula in further detail, to include live classroom instruction, to see how we might apply our Phase II empirical efforts on an even wider scale.

Use of a Design Pattern Methodology

Design patterns are components of an overall system that can be abstracted and generalized (i.e., reused) to other domains. The "system" can be any finished product that is delivered to a user, such as a computer display, a functioning piece of software, or even a game. It is important to note that we define design patterns not merely as physical design layouts that can be applied from one content domain to another. Rather design patterns are a combination of (1) the display format of the human computer interface (HCI) at the presentation level, (2) a set of procedures or principles that govern functionality and application (akin to the business logic discussed in software design), and (3) an underlying database which organizes system information and links the HCI aspect of the design pattern to other, hidden elements of the system (Spiker and Dick, 2007). The design pattern concept originated in the fields of architecture and computer science, where the application of these well-tested, reusable “templates” not only helps designers select appropriate solution methods for a given design problem more efficiently, but also significantly reduces the time and cost of developing new products (Spiker and Dick, 2007).

As applied to serious games, the design pattern methodology involves identifying proven game element components that can be successfully “migrated” from a training game in one domain and applied to the development of a training game in another domain. Thus, the design pattern methodology is particularly well-suited for aiding in the development and utilization of the TARGET tool.

The following illustrates how the principles underlying the design pattern methodology was applied in creation of the FMS game.

The FMS game employs two main design meta-patterns, each of which is composed of multiple component data patterns. One is the main FMS Programming Game, and the other is an embedded game called the Pilot’s Lounge, which players can access between levels of the main game. Below is a detailed summary of the main features of the game, framed in terms of the three design pattern elements (presentation, business logic, and database integration) described in Spiker and Dick (2007).

Design Meta-Pattern #1: The “Main Game”

The “Main Game” provides whole-task training and takes place in the simulated cockpit of a CRJ Jet. It provides players with a context-rich, realistic environment where they assume the role of the first officer who must program the FMS preflight sequence, update the FMS as new information arises, detect errors and solve problems, all under the guidance of a “virtual captain” who provides prompts and feedback.

The main FMS game is composed of several component design patterns, some of which are illustrated in the annotated snapshot of the game, shown in Figure 16. These include:

Figure 16. FMS Main Game interface.

1) **A game control toolbar** which provides help and allows players to repeat audio, view audio as text, view scores, and set other options.

2) **A realistic interactive device** (here, the FMS) with an underlying database that corresponds to real-world function.

3) **A countdown timer** that corresponds to actual time constraints on the flight line.

4) **Communication** with additional agents (here, via intercom) that introduce interruptions, new information, and challenges.

5) **Task sequencing** (steps) that emulates real-world process and procedures (and a visual tracker of step completion).

6) **Supporting materials**, such as charts and memos that correspond to real-word material that must be integrated into the task.

7) **Task stressing display** (here, the “taxiway”) that shows the aircraft’s position in the taxiway lineup

8) **Cognitive apprenticeship** in the form of a virtual captain who provides verbal branching prompts and tailored feedback using scaffolding logic. This is described in more detail in Table 12.

Below, we provide an overview description of the main FMS game, organized around the three design pattern elements: presentation, procedures and principles, and database integration. Space constraints preclude describing each constituent design pattern, so only select ones are highlighted.

1) Presentation (Display Content and Format)

Embedded in the cockpit panel are various instruments and tools the player can enlarge and interact with, such as the FMS itself, the multifunction display (MFD), primary flight display (PFD), flight control panel (FCP), and other instruments related to FMS functioning, as well as a clipboard with forms and charts that players will need while programming the FMS. (See Figure 16 for a snapshot of the FMS Game interface.) To the right of the cockpit is a bar that depicts visual cues informing players of their performance, as well as an intercom that gives players periodic contact with external entities (e.g., ATC, weather-ATIS).

2) Procedures and Principles ("Business Logic")

The player assumes the role of First Officer (the one primarily responsible for programming the FMS). Seated next to the player in the cockpit is a virtual captain. The captain interacts with the player in several ways, including verbally, by pointing to certain features on the FMS, and in writing, at the end of a session, giving them memos summarizing their performance. The virtual captain acts as a scaffold for the player, allowing the latter to serve in a cognitive apprenticeship role, as the former provides prompts and hints, gives local and global feedback (i.e., immediate and end-of-session summary), administers some of the positive and negative consequences, and models procedures and problem solving (particularly at the earlier levels). The prompts and feedback resemble natural language ("pilot talk"), with affective reactions (e.g., tone of voice) corresponding to how well the player is doing. Table 12 provides a template of the interaction between the player and virtual captain. If the player is able to do a task on his or her own within, say 10 seconds of the captain asking "What's next?" then the player receives positive feedback, and a vague prompt ("What's next?") for the next step. If the player takes too long (e.g., more than 10 seconds), or takes the wrong action, the captain then goes to the second, more explicit prompt, and so on.

Table 12. Scaffolding and Feedback Design Pattern
(with examples from Level 1, Introduction to Preflight Programming)

Time (seconds)	**Prompt (Captain Says):**	**Feedback (Captain Says):**
0-10	"Okay, what' the first step?	Nice job! You remembered the first step. We'll have this thing programmed in no time.
11-20	(pause) "First, we have to check that the database is active". [STEP TO TAKE]	Good, you remembered what page it's on
21-30	(pause) "That's on the status page" [WHAT PAGE TO GO TO CARRY OUT THE STEP]	Okay, so maybe you didn't know the step, but you did know how to get there
31-40	(pause) "You can get there by pressing the line select key next to "status" on the "index" page menu" [ACTION TO TAKE TO GET TO THE PAGE]	Good, but try to find it on your own next time
41-50	(pause + annoyed tone) Come on, you get to the status page by hitting this line select key next to the STATUS option (finger points to correct buttons FMS panel). [PHYSICALLY POINTING TO CORRECT BUTTONS]	Let's pick up the pace or we'll never get out of here.
NOTE: "Time" starts when captain first says "Okay" If the player gets to the page within the time range, the captain says the corresponding feedback line. If the player does not get to the page within the time range, he or she gets the next prompt		

Via the intercom and radio, the player also interacts with other, external "agents" such as flight attendants, air traffic controllers, and others who present the player with interruptions, problems, and unexpected events

that the player must handle. There are six levels of the game that progress with increasingly complex problem-solving challenges, greater time pressure, and less guidance from the virtual captain. Players are scored based on time, accuracy, and the amount of help (from the virtual captain) needed.

3) Database Integration

Importantly, the "game" FMS behaves and functions like an actual FMS, mimicking the keystrokes and displays. So too, the functionality of the FMS game reflects many of the behaviors, situations, and constraints of a real-world cockpit environment. Thus, players experience the realistic consequences of their choices and actions, such as angering the captain and being diverted back to the gate because their preflight programming was not completed in time, or having their plane fly off route due to data errors. They are also faced with realistic challenges, such as having to update the FMS in response to a diverted route. Other elements of the game, such as the amount of time allocated to the various tasks, the content of the feedback, and the functioning of the devices and supporting material, are organized and linked with other elements within the system in order to present players with a coherent, compelling game-playing experience.

Design Meta-Pattern #2: The Pilot's Lounge

The Pilot lounge environment itself, along with each of the arcade gamelets contains a number of distinctive design patterns that could be applied to other domains. For example, some of the display features of the games could be swapped out to address domains that share common elements with FMS operation. Thus, we could substitute torpedoes for bombs, an inventory control device for the FMS, and a conveyor belt conveying parts for the circling spy planes. Using this replacement logic would allow the gamelets to be readily modified and adapted to a host of similar tactical problem solving settings.

Migrating Serious Game Design Patterns to Other Domains

When we began this project, we acknowledged that a main issue in serious game design is that it is often impractical and expensive to design, from scratch, games that have a narrow focus and limited application. Furthermore, there has been a fairly recent mandate within the military to compile collective standards and specifications (referred to as the Shareable Content Object Reference Model, or "SCORM") so that the content of web-based learning materials may be re-used and exported to other training programs – something that can be particularly challenging with game-based training (Conkey, Smith, DuBuc & Smith, 2006). With that goal in mind, we developed the specifications for our FMS Programming Game in accordance with the principles of design patterns. With this methodology, we ensured that the underlying structure and component design patterns could apply to a variety of similar training environments, such as systems management, operating a shipboard navigational computer, and even working with complex machine shop tools and devices. These modular design patterns allow the subject matter of the FMS game to be swapped out with that of other training areas. For example, the FMS can be replaced by the interface of another type of programming device (e.g., inventory control), and the virtual airline captain could be substituted for other virtual characters, such as a master crew member or master machinist, who provide appropriate scaffolding and cognitive apprenticeship. Yet, the overall structure of the game, the general approach, the successive prompts, the cognitive apprenticeship, the types of pressures, etc., would still form a valuable foundation upon which a new game could be adapted.

Along these lines, Table 13 illustrates how the reusable design patterns developed for the FMS game might be migrated to develop serious games for two other training areas. The rows of the table correspond to the three design pattern elements of presentation, business logic, and database. The second column provides an abstract description of 26 components that make up our palette of games. Column three provides examples of how these were employed in the FMS programming game. Columns four and five give examples of how these can be applied to two different games: (1) one designed to train navigators how to program navigational equipment on the Navy's amphibious landing craft air cushioned (LCAC) vehicle, and (2) the other designed to train precision machinists how to program the CNC (Computer Numerical Control) device used in the fabrication of high-tolerance metal parts. These examples illustrate how the principles of design patterns can be used to improve the effectiveness and efficiency of developing serious games for training while realizing substantial returns on investment (ROI).

Table 13. Framework for Applying FMS Serious Game Design Patterns to Other Domains.

Design Pattern Features	Abstract Design	Design Applied to FMS	Design Applied to Navy LCAC – Navig.	Design Applied to CNC
1) Presentation Display Format	a) main environment	a) cockpit	a) LCAC cabin	a) machine shop
	b) the device	b) FMS device	b) nav. equipment	b) CNC interface
	c) supporting materials (charts, data, equipment)	c) clipboard w/ charts, dispatch + MFD, map display, etc.	c) charts, tools gps display, radar, etc	c) blueprints, material specs, notes from engineer, etc.
	d) virtual mentor	d) virtual captain	d) craftmaster	d) master machinist
	e) goal marking device: timer	e) scheduled takeoff timer	e) H-hour countdown	e) countdown of project due date
	f) communication device	f) intercom/ incoming messages	f) Radio, intercom, other crew	f) telephone, cell, computer
	g) outside communicators	g) Air traffic control, flight attendant	g) engineer, deckmaster, etc.	g) engineers, co-workers, customer
	h) progress toward goal tracker	h) preflight steps checklist	h) location checkpoints	h) parts completion checklist
	i) time pressure device	i) taxi way – plane approach takeoff	i) support ship begins to move away	i) customer or engineer walking toward shop
	j) performance visualization	j) plane's trajectory (higher levels)	j) distance from route; trajectory	j) simulated parts
	k) standardized top tool bar	k) standardized top tool bar	k) standardized top tool bar	k) standardized top tool bar
	l) lounge room	l) pilot's lounge	l) LCAC lounge	l) break room
2) Procedures and Principles ("Business Logic")	m) player assumes role	m) First Officer	m) LCAC Navigator	m) Machinist
	n) cognitive/behavioral realism	n) ex: following FMS preflight procedure	n) ex: following LCAC procedures	n) ex: following actual CNC procedure
	o) scenario-based exercises	o) ex: "Storm requires new route"	o) ex: "Rudder stuck, go off cushion"	o) ex: "Engineer changed plans"
	p) whole-task training and scaffolding	p) ex: first level, player navigates, captain inputs data	p) ex: first level, player identifies location; master does computation	p) ex: first level, player inputs cut depth, master inputs coordinates
	q) cognitive apprenticeship	q) captain provides guidance	q) craftmaster/others provides guidance	q) master provides guidance
	r) successive branching prompts	r) ex: first prompt is the step, second the page it's on,	r) ex: first prompt is to check X & Y, second is how	r) ex: first prompt is the step, second how to get there
	s) tailored branching feedback	s) ex: If no hints, feedback = "Great job" ; if many hints, feedback = "Need to learn that step"	s) ex: If no hints, feedback = "Great job" ; if many hints, feedback = "Need to learn that step"	s) ex: If no hints, feedback = "Great job" ; if many hints, feedback = "Need to learn that step"
	t) multiple challenge levels	t) increase in number of tasks, problem types, and complexity	t) non-north orientation; equipment failure frequency	t) increase in complexity of design, number of inputs
	u) audio/visual input	u) Ambient cockpit sounds, lights	u) Ambient ship sounds, lights	u) Ambient machine noises, lights
	v) Multitasking challenges (e.g. interruptions)	v) Flight attendant requests attention, etc.	v) Radar goes out, other ships in area, etc.	v) New rush job, tool breaks, etc.

Table 13 (Continued). Framework for Applying FMS Serious Game Design Patterns to Other Domains.

Design Pattern Features	Abstract Design	Design Applied to FMS	Design Applied to Navy LCAC – Navig.	Design Applied to CNC
3) Database Integration	w) Realistic consequences x) Realistic devices y) Realistic prompts & feedback script z) Realistic time and other constraints	w) FMS data input affects flight path x) FMS buttons work y) Realistic prompts & feedback script z) Five minutes to program Preflight	w) Navigation inputs affect ship course x) GPS updates y) Realistic prompts & feedback script z) At location within 5-mins of schedule	w) CNC inputs affect how part is cut x) CNC cuts/drills y) Realistic prompts & feedback script z) Set parameters for different metals

Identification of Navy Claimants

Because Anacapa has a variety of other active R&D programs with the Navy, we have been able to keep our technical points of contact abreast of our developments in GBT stemming from the present project. Several of these individuals have expressed keen interest in our work, as they are presently experiencing training-based problems for which a GBT solution holds promise. Specifically, we summarize below the contact information for four Navy claimants, along with a description of their present needs, the current approach being used in their domain, and how they might use our GBT products. Collectively, these claimants represent a very diverse group of platforms, combat areas, and technical issues. However, all have serious problems involving training and all are quite interested in incorporating GBT into their training curricula. A version of this descriptive summary is also presented in the Transition Plan. For completeness, we also provide a comparable summary of a potential private sector claimant, Mesa Air, whose pilot development office is co-located with ASU-AMT.

Navy – NAVSEA – DDG 1000

- **Defined needs & operational gaps**: Dramatically reduced manning is a Congressionally mandated requirement for late-stage program funding. Based on analyses to date, cross-specialty training and job/rate consolidation are essential to achieving the needed level of manning reduction. Adaptable on-the-job training is likely crucial to success of the first flight of ships in this high-risk departure from traditional manning redundancy.
- **Current approach vice benefits from our approach**: The DDG 1000 Program has performed dozens of man-years of job analysis and job restructuring studies, yet doubts remain about the workability of reduced manning for this platform. To the extent that reduced manning is attempted for this platform, Anacapa's *GBT2*[1] will (a) support the requirement for adaptable on-the-job training; (b) support cross-discipline training/rate consolidation; (c) elevate levels total-crew performance; and (d) mitigate operational risk associated with reduced-manning.
- **Tests and demonstrations for operational acceptance**: Use Case A: Many instances of job-specialty consolidation under consideration in the DDG 1000 Program involve necessary-but-non-critical KSAs where the benefits of Anacapa's *GBT2* are obvious (e.g., the planned consolidation of store-keeping, laundry, and mess assistance specialties). In these cases, testing of *GBT2* curriculums will not require rigorous control-vice-experimental group testing prior to operational acceptance. SME evaluation will suffice, mainly to verify that the cost of any given curriculum is worth the investment.

  Use Case B: There are other types of job-specialty consolidation contemplated for DDG 1000, however, where hyper-critical KSAs absolutely cannot be compromised. In these cases, the onus is

[1] GBT2 stands for game based training technology, which is a combination of our GBT approach being described here coupled with the use Design Patterns to facilitate re-use of concepts and software. This innovative methodology is described later, in Chapter 7.

on the DDG 1000 program itself – particularly the design agent for each suite of systems – to prove that these KSAs can be consolidated in a smaller number of team members than done traditionally. The onus, likewise, is on the program to create, test, and prove the training schema for supporting such consolidation. From both the design agent's and Navy's perspective's, the decision to augment formal classroom and simulator training with *GBT2* is straightforward: The marginal costs of *GBT2* "augmenting curriculums" pales in comparison to the potential cost of not doing everything possible to maintain the hyper-critical skill sets.

- **Example Intersection with *GBT2***: Mr. Paul Matthias, Technical Director for DDG 1000 Human Computer Interface Design, Raytheon Company [2]. Our prior relationship with Mr. Matthias' engineering group [supporting test design for the Integrated Undersea Warfare (IUSW) system EDM] and our current relationship with his group [in a Phase II SBIR to apply our proprietary integrated Tactical User Interface (*iTUI*™) technology to the IUSW HCI] provides a channel to the IUSW training support group. We are positioned to provide a compelling application of *GBT2* to enhance/maintain several types of hyper-critical skills in ASW.

B. Navy – PEO SUBS – Control Room aboard Virginia Class Submarines

- **Defined needs & operational gaps**: Shared situational understanding (SSU) is a hyper-critical team skill in submarine operations, especially for the Approach Officer[3], Sonar watch, and Fire Control watch during target engagement. Lapses in SSU – even in peace-time ops (e.g., Greeneville incident, 9 February 2001: problems in SSU between CO and ST) – can be catastrophic.
- **Current approach vice benefits from our approach**: Structured opportunities to build skills in Control Room SSU largely occur in simulators, which are highly constrained resources. Structured maintenance and enhancement of these hyper-critical skills are therefore infrequent. *GBT2*, however, can capitalize on the fact that off-duty submariners on deployment are a "captured audience", amenable to entertainment, even if it were game-based training. Second-generation *GBT2*, employing multi-player games, would increase the Control Room team's opportunities for structured SSU skill building and maintenance by several orders of magnitude.
- **Tests and demonstrations for operational acceptance**: As in the case of DDG 1000, above, SME evaluations would likely suffice for operational acceptance. Again, the costs for SSU-enhancing game curriculums are miniscule compared to the costs of not providing this technology.
- **Example Intersection with *GBT2***: Mr. Jack Griffin, SBIR/STTR coordinator at Naval Undersea Warfare Center (NUWC) Newport. We are working with Mr. Griffin on the Phase II SBIR *iTUI* application mentioned above re DDG 1000. He will be providing us a channel to PEO SUBS for Phase III transition of *iTUI*. As in the case of the IUSW system aboard DDG 1000, we will be ideally positioned to provide compelling application of *GBT2* for the submarine community.

C. Navy – NAVSEA – Surface Electronic Warfare Improvement Program (SEWIP)

- **Defined needs & operational gaps**: Lack of hyper-critical skills in operation of the SLQ-32 Electronic Warfare System was a direct contributor to loss of life in the 1987 USS Stark incident, 37 fatalities. Lack of SSU, especially between the TAO and EW watch, contributed to the 1988 Vincennes incident, 290 fatalities.
- **Current approach vice benefits from our approach**: Re-design of the HCI is part of the Navy's long-term solution to the SLQ-32 problem. In fact, our *iTUI* technology was founded through research to develop this solution, which yielded dramatic HCI innovations judged by Navy evaluation

[2] Raytheon Company – Integrated Defense Systems Division – is the prime mission systems equipment integrator for all electronic and combat systems for the DDG 1000 Zumwalt Class Destroyer program.

[3] Which is often, but not necessarily, the CO.

of a full-functioning prototype to reduce the EW watch by 50% while simultaneously improving watchstanding performance. Even with an improved HCI, however, over-learning hyper-critical skills in contact identification is crucial to effective missile defense, especially given the combined effects of increasing littoral ops and continuing proliferation of relatively inexpensive, shore- and small-craft-fired missiles. *GBT2* is an ideal means for providing such over-learning.

- **Tests and demonstrations for operational acceptance**: The same arguments apply here as in the cases of DDG 1000 and attack submarine Control Rooms.
- **Example Intersection with *GBT2***: Mr. Thomas Hayes, Program Management for SEWIP, General Dynamics Advanced Information Systems Division. We worked extensively with Mr. Hayes in developing the original *iTUI* solution to the SLQ̃32 HCI problem. He has agreed to collaborate with us on application of *iTUI* technology to SEWIP. Again, this provides us a direct channel to downstream SEWIP training development and positions us for a compelling *GBT2* application.

D. Navy – Expeditionary Warfare Training Group Pacific – Landing Craft Air Cushion (LCAC)

- **Defined needs & operational gaps**: Attrition during the simulator training phase for LCAC trainees is a chronic problem that has manning implications downstream. The training curriculum is demanding, particularly in areas of multi-tasking, where Navigator and Craftsmaster crew positions suffer from trainee dropout and washout as training proceeds. There is both a manning loss as well as a major cost impact from resource expenditures on trainees who do not finish the program.
- **Current approach vice benefits from our approach**: Training presently consists of classroom lecture followed by simulator training on individual and crew tasks. There is no opportunity for self-paced “spin up” or “remedial” training as trainees with specific deficiencies in multi-tasking skills are identified.
- **Tests and demonstrations for operational acceptance**: EWTGP is receptive to using an LCAC serious game (patterned after a combination of the FMS and callout-procedure games) for several consecutive classes at their Camp Pendleton training site. Both user reaction data (to the game) and comparison of in-simulator performance to non-game (control) groups of trainees will be collected.
- **Example Intersection with *GBT2***: QMC (SW) Brady G. Annas. Expeditionary Warfare Training Group, Pacific N71, LCAC Navigator Training. COMM 619-437-5681 DSN 577-5681 brady.annas@navy.mil. Commitment to evaluation of *GBT2 intermediate* products as Phase II progresses.

E. Private Sector – Mesa Air

- **Defined needs & operational gaps**: Like ASU-AMT, Mesa Air has prescribed needs for improving and accelerating FMS and procedure-callout proficiency among its beginning line pilots. New pilots who are weak in these skills (as first officers) pose greater time demands on senior captains who must assume greater responsibility for FMS and callouts than their standard operating procedures originally intended. This in turn creates greater safety risks and impacts an aircraft’s ability to meet its target takeoff time and desired fuel efficiency, both having significant cost impacts.
- **Current approach vice benefits from our approach**: Present deficiencies in core FMS and procedure-callout competencies are handled by the senior pilots taking more responsibility in the cockpit for these duties—a safety issue—as well as expending “off clock” time for remediation. This has both morale and cost implications. Mesa Air does not presently have other means to accelerate or remediate skill progression for these competencies.
- **Tests and demonstrations for operational acceptance**: Mesa Air will have a subset of its beginning pilots experience the FMS and procedure-callout games in the course of their current training. Their performance in Level D (high-fidelity) simulator training on these core competencies will be compared to a control group of beginning pilots who do not experience the games.

Example Intersection with *GBT2*: Mr. Pete Hayes, President Mesa Pilot Development, Mesa Airlines. He has expressed an extremely high degree of interest in, and support to, the Phase I research and proof of concept. He has committed to carefully evaluating Phase II products and is a "high-value target" for commercial transition.

Chapter 8: Experimental Design

A driving force behind the original solicitation was the recognized need for unequivocal demonstrations of the effectiveness of game based training (GBT) to promote learning, retention, and transfer of training. The paucity of extant data on the ability of serious games to promote training was clearly articulated by Hays (2005) and has been a frequent point of issue in other contexts, most notably the Serious Game ListServe. To that end, a major research product in Phase II will be the conduct and analysis of a tightly-controlled experiment in which the impact of a functional serious game (our FMS Programming Game) is compared to conventional training. A complete specification of the experiment (i.e., number of subjects, specific measures, locations, ASU-Mesa classes, timeframe, schedule, etc.) will be described in a Phase II proposal, should Anacapa be asked to submit one. However, we want to describe the logic of our evaluation design in this document so the government can see how we plan to exploit the unique capabilities that ASU-Mesa offers as a testbed for serious game implementation and evaluation. Portions of this plan were described in our previous interim reports and were presented at project meetings. Our thinking on the requirements of the evaluation study has matured since then and is described below.

Conventional Training – Background. Because the focus of our evaluation is an assessment of GBT relative to conventional training, we must at this point offer a brief, but useful, digression in which we describe the context in which ASU-Mesa students are presently trained in FMS operating skills. This description serves several purposes. On the one hand, it is necessary to appreciate the richness and complexity of instruction that goes on in any operational training environment, with ASU-Mesa being no exception. On the other hand, and even more important, this discussion is offered here so the reader may appreciate that students receive several, repeated exposures to FMS training. Hence, our GBT functional prototype will be implemented in the context of a considerable amount of training devoted to this very important piece of avionics. As the discussion will hopefully illuminate, while this FMS training is content-rich and very advanced, there are still problems concerning lack of self-directed practice and inability to gain the automated, repetitive practice needed on key FMS skills so that the student's capstone simulator training is more effective. Introduction of serious GBT into the ASU-Mesa curriculum is designed to solve this problem.

One of our key Phase I activities was to identify and delineate the current (conventional) methods that ASU-Mesa presently uses to teach student-pilots FMS operation. Discussions with ASU-Mesa faculty revealed that students receive virtually no exposure to FMS operations during the first 2 1/2 years of their undergraduate training (i.e., through the first semester of their junior year). This is because their aviation training primarily involves acquiring the flight experience needed to obtain a private pilot's license, a certified flight instructor's (CFI) license, a commercial rating, an instrument rating, a certified flight instructor instrument (CFII) rating, and a multi-engine rating. None of the aircraft associated with these licenses and ratings—the PA28-161 Piper Warrior, F-36 Beechcraft Bonanza, and B-58 Beechcraft Baron—have an FMS as an onboard avionics systems.

During the second semester of the student's Junior year, they receive coursework in aircraft systems for the regional jet (AMT 482). This is their first exposure to the FMS, along with other more complex avionics systems. The bulk of this training involves lectures in which PowerPoint slide presentations are provided that describe the major modes of operating the FMS, how the FMS can be used to enhance aircraft performance (fuel efficiency, airspeed control), and how the FMS can improve navigation accuracy. Actual hands-on instruction with the FMS does not occur until the beginning of the student's Senior year, when as part their Airline Instrument Procedures and Air Navigation classes, they have access to a computerized simulation of the FMS through the Aerosim Virtual Flight Deck (VFD™). The VFD™ simulation is provided as a set of self-directed student exercises performed as homework in the department's Learning Laboratory, where the simulation program is resident on each PC workstation in the laboratory. The Airline Instruments and Air Navigation classes are vital to the student's learning and understanding of how the major avionics systems work in regional jets and instrument-based navigation is the backbone of commercial flying. Unfortunately, the King Air simulator (see Figure 17) which provides the students with hands-on practice to supplement

classroom training, does not have an FMS as part of its avionics suite. Consequently, the student's main opportunity to receive hands-on experience with the FMS is through the VFD™.

Figure 17. King Air simulator.

The aforementioned summary encapsulates the ASU-Mesa student's training in FMS operations until they begin their capstone CRJ flight training device (FTD) class during the last semester of their Senior year. The FTD provides a very high fidelity representation of the CRJ 200 cockpit, including a fully functioning FMS. As was noted in the first interim report, if the students have not engaged in sufficient self-directed study and practice with the VFD™, they will have difficulty performing the core tasks required to operate the FMS accurately and efficiently. These include initializing, programming, troubleshooting, and error-correcting.

In recent visits to ASU-Mesa, we observed some additional training exposure to FMS operation now being given to students, and which must be incorporated into the baseline condition when we assess the effectiveness of GBT. Specifically, the instructor for the Airline Instrument Procedure class is now having students select some very specific aspect of FMS operation, such as shooting a missed approach or being vectored near the terminal area, and then use the VFD™ to create a 15-minute PowerPoint class presentation in which pictorial representations of modes, sequences, and critical conditions on that topic are displayed and explained to the other students. This assignment gives the students a more practical and structured basis for using the VFD™ , and it appears that, at least for the assigned topic, it will give the student some knowledge and practice they otherwise would not have. This pedagogical approach is likely to continue, and must be accounted for in the conventional training we designate as our "control" condition during the GBT evaluation. Far from being a complication, this additional conventional training will actually given us an additional measure of performance in our FMS GBT evaluation study, as discussed below.

Track One – Overall Impact of Serious Game-Based Training. In describing the logic of our evaluation design, the first point is that we plan to use a two-track approach for assessing GBT. Track One will assess the overall effectiveness of serious GBT compared to conventional training. This assessment will be restricted to ASU-Mesa students who are entering the regional jet phase of their training, where FMS operation is being introduced. Within a given year, including the summer, this will provide us with approximately 50 students for testing. With 25 subjects per group, this would be a sufficiently large sample size to detect meaningful differences between game-based and conventional training should any effects be present.

Subjects would be randomly assigned to the GBT and control conditions. It is important to note that this will be a very tightly controlled comparison in that the students in both conditions would receive the same classes, labs, practical exercises, access to VFD™, and all the other ASU-Mesa supplied experiences with FMS operation (which encompasses 3 different types of FMS) that were described above. Our close working relationship with ASU-Mesa has allowed us to receive the freedom (and trust) to manipulate their on-going

training program for the purposes of this research; this access is control is vital, and we believe it will be a major determinant in achieving the highly-controlled experimental manipulations discussed herein.

As a result of this control, the only difference will be in the type of web-based FMS training package to which students are given access. The control condition subjects will be given a url and password that allows them access to the ASI-developed interactive FMS trainer that will be created. Indeed, we plan to create the entire training package the control group receives so that we may ensure that it has all the information and simulation capabilities present in the gamed-up version, absent the gaming elements. Otherwise, it is easy to confound improved training approaches (e.g., interactivity, feedback, visuals) that are not gaming elements per se. Access will be provided to students during the first month of the second semester of their Junior year.

The treatment group will be given another url and a password so they may access the gamed-up FMS trainer. This web-based training will contain the same information content, progression of lessons, and visual format as the above, only it will have a game-based "wrapper" that the control condition does not receive. As described in Chapter 5, this includes the situation realism (storyline scenario, the "virtual" captain, time pressure, airplane icons on the taxi-way, etc.) and challenge (points tallied and reported, relative rank with other students, consequences of failure, etc.) dimensions that we believe make games unique relative to conventional training. It will also include the embedded Pilot's Lounge arcade gamelets that promote repetitive part-task practice on some core problem-prone skills. Importantly, we will collect comparable process measures on the two conditions—amount of time spent in self-directed activity, reaction to the experience, interest level in receiving more, problems understanding material, etc.—as well as the extent to which learning/training with the system (control vs. GBT) transfers to other settings.

For this transfer, we have two possibilities that we refer to as "near transfer" and "far transfer" respectively. Near transfer entails measuring post-training performance in some setting that resembles the training setting, such as comprehension of some other FMS simulation experience. The most logical choice here would be to give each subject, following their FMS training, some exercises to perform on the VFD™ and see how they do. As discussed above, the recently-added lab exercises with the VFD™ will provide a convenient vehicle for measurement that is now a normal part of their training flow. Far transfer would entail measuring performance in some setting that is more job-like. For this assessment, we would measure how each student performs on the real-life FMS that resides in the CRJ FTD. This assessment would be done during their 10 capstone lessons (second semester Senior) in the FTD. We would take periodic measures of FMS performance (including amount of instructor intervention required to remediate and progress students) throughout these lessons. A host of detailed measures (time to program, nature and number of errors) would be collected over some or all of these capstone lessons.

Depending on the student throughput, we could also follow the student's progression to Mesa Air (about 80% of ASU-Mesa's graduates take their initial employment with Mesa Air). As part of a student's introductory phase-in training, they receive sessions in the airline's very high-fidelity Level D simulator, in which the FMS must be programmed just as they do on the flightline. Opportunities to collect additional far-transfer data will be available, and Mesa Air has agreed to let us collect these data.

Besides the process and performance measures, we will also collect what we would call "implementation" measures during the evaluation. By implementation, we mean the personnel, equipment, and time resources needed to implement each condition. This will include the costs associated with creating the training, hosting it on the web, maintaining the site, and so forth. It will also include the instructor and engineering costs associated with measuring performance in the near-transfer and far-transfer tasks. By keeping track of the implementation resources, we will be able to perform a comprehensive return on investment (ROI) analysis to precisely determine the cost benefits associated with GBT. In this regard, ASU-Mesa has a very detailed cost model of simulator time, instructor costs, and material costs associated with producing students who are "airline worthy" upon graduation. We will capitalize on this framework in conducting our own ROI analysis. For example, we know that the FTD costs $125 to operate, with instructor costs at $50/hour. The costs to remediate a student with an additional 30 minutes because they do poorly on the FMS—which we have observed first hand on more than one occasion—are thus $87.50. While this does not seem like much, if one multiplies that amount over 5-10 lessons over 10-20 students, and that begins to add up. Also, this is only one place where deficiencies in FMS competencies show up. There are others. For instance, even

more costly are the Level D simulators noted above, costing $500/hour to operate, where problems with FMS operation also appear. These and other areas of implementation will be carefully scrutinized and incorporated into a detailed ROI for GBT.

In sum, we believe this plan offers the “cleanest” possible comparison between game-based and non-game based training since the control group receives virtually the same training as the GBT except for what we might call the “gaming wrapper.” For this test, we will put ALL the gaming elements from the situational realism and challenge dimensions into the “gaming wrapper” to see if there is a significant increase in either near-transfer or far-transfer relative to the control. If there is, then further work is needed to explore *which* elements are responsible for this effect so that we may offer the most efficient application strategy possible to other job domains. That is the purpose of the second track described below.

Track Two – Effect of Individual Gaming Elements. The purpose of the second track is to identify, at a more focused level, where GBT effects may be showing up. This will involve conducting several follow-up studies in which we selectively remove one or more of the gaming elements that reside in the “full-up” SGBT version, described above, to create a “gamed-down” trainer that can be compared to more conventional training. For Track Two, we plan to use a combination of less experienced student pilots, Mesa Air line pilots, and non-pilots to make our assessments. Consequently, most of our testing will be done by means of a near-transfer setting, in which subjects will receive post-training assessment as a combination of comprehension questions, problem-solving exercises, and low- to moderate-fidelity PC-based FMS simulations. When Mesa pilots are used in one of these studies, we will have the added ability to collect “far-transfer” data in their Level D simulator.

To create the gamed-down version, we might elect to remove the ambient stimulation (e.g., airplane sounds), auditory feedback, and the challenge (e.g., point tallying) in the GBT for the experimental group and then compare that to the basic FMS trainer. Other manipulations might include reducing the complexity of the feedback in the game version (we presently have a five-layered feedback prompts based on student performance), eliminating the Pilot’s Lounge embedded gamelets, replacing the virtual captain with some text-oriented prompting, or reducing/eliminating the time pressure. Regardless of the manipulation, both systems (gamed-down trainer vs. conventional trainer) would be web-based as described above, with students given self-directed exercises that are to be completed. Our subject pool would consist of Freshmen and Sophomores at ASU-Mesa, students at our local city college (Santa Barbara City College), students from our local university (University of California Santa Barbara, UCSB), and Mesa Air line pilots who pass through ASU-Mesa for periodic refresher training. This latter group has recently been made available to us by the kind permission of the president of Mesa Air Pilot Development (Mr. Pete Hayes). Mesa Air has expressed interest in the potential of using serious game technology to enhance training, and they will be monitoring the results of our study with special interest.

We have been highly successful in recruiting subjects from all four sites as part of a multi-tasking project we have been conducting for ONR. In the past year, we have collected web-based testing data from 300 UCSB students and 400 SBCC students, so we have a proven track record in recruiting subjects and administering tests. At the same time, we have collected web-based training and flight scenario data from 60 ASU-Mesa students in little over a month. We have been recruiting Mesa Air pilots to support several surveys of scenario based training in another project we conducting for NAVAIR Orlando. We also have human subject committee approval for collecting data at both sites that will be in place for the coming year. Hence, we believe we have a proven track record of subject recruitment and test administration that will be essential for conducting the necessary Track Two tests. Over the course of a Phase II project, it is likely that several studies would be needed to test the impact of different combinations of gaming variables on training effectiveness. In the aforementioned multi-tasking study, we were able to conduct six separate studies using the subject populations described above. We would see no problem achieving a similar target figure in the GBT project.

Utilization of Research Findings. As the evaluation studies are conducted and turned into research products, we will utilize the findings in several ways. One, they will be used to guide the design of our second functional GBT, the callout-procedure game. Two, the experimental results will be used to populate the TARGET tool where our research data will be combined with the literature review and Web surveys to create an enriched database of game element-training environment relationships. Three, the results will be

reported in peer-reviewed journals and at professional conferences (e.g., Software Engineering Research and Practice, Interservice/Industry Training and Systems Equipment Conference, Human Factors and Ergonomics Society, Serious Game Summit).

Chapter 9: Additional Feasibility Issues

Another important task we specifically targeted during Phase I was to ensure that we have the in-house capabilities to successfully complete the required technical work required for a Phase II effort. To that end, we conducted two specific activities to ensure that our staffing mix would be sufficient to meet the project needs. Specifically, we (1) reviewed the feasibility of our projected game development activities with our in-house programmer and (2) arranged to obtain the services of an experienced game designer. Regarding the former, we discussed the proposed game design implementation with our staff programmer, Mr. Evan McPeters. Mr. McPeters has extensive experience designing interactive hmtl and flash-based training programs. In this regard, Mr. McPeters has supplied the software expertise to support the creation of our SCORM-compliant critical thinking training program for the Army Research Institute. Termed $(CT)^2$ (for Critical Thinking Computerized Training). This program makes extensive use of all the enabling technologies that will be needed for our FMS programming game, such as animation, sound, high levels of interactivity, bit-intensive graphics, timing, score recording, and the like. It also supports links to other learning management systems should that be needed. From our discussions, Mr. McPeters has indicated that all the proposed and assumed capabilities in the FMS storyboard described in the present report are very feasible from a programming perspective, and can be accommodated within his existing skill set.

Importantly, Mr. McPeters is also experienced in creating team training programs, as reflected in our six-person team trainer entitled SamePage. This scenario-based, small-unit training system was developed for ARI and its software basis was entirely the creation of Mr. McPeters. Through use of a client-server architecture, the system runs on the public Internet with minimal lag times owing to an inventive combination of a database-driven scenario controller, scenario timer and master clock, shared document server, and specialized networking features. The latter included the use of remote scripting and XMLHTTPRequest object manipulation to control the exchange of client-server information transmissions in an efficient, timely fashion. This experience will be especially valuable if we elect to have students compete with one another directly, where FMS programming tasks are controlled via a centralized server.

The final addition to our project team has been the agreement of Mr. Robert Karp to serve as our game design consultant. Mr. Karp has extensive experience in both designing games and is an avid game player himself. He was one of the main architects of the Warcraft 3 game, and is an experienced player and designer of MMOGs. In addition to game design, Mr. Karp has considerable experience in creating multimedia products for the entertainment industry, with particular expertise in video editing, sound recording and editing, photography, and lighting. This experience has given him a wealth of exposure to games of all types, so that his intuitions of what makes games interesting, fun, challenging, and so forth are informed with knowledge of best practices and state of the art game design. Located in the Phoenix area, Mr. Karp will be able to view our products at the ASU-Mesa testbed, simplifying the logistics for meetings and face-to-face consulting. An in-depth resume for Mr. Karp will be provided in a Phase II proposal submittal.

Chapter 10: Conclusions and Recommendations

The overall goal of this project is to develop a means to determine, empirically, whether game-based training results in better training effectiveness compared to conventional training, and, if so, which game elements and game element combinations are most effective for a particular training environment. Based on our Phase I work we offer the following conclusions and recommendations:

1) Conduct a systematic task analysis of the training needs and objectives of the target training environment. This was demonstrated in Task 1, where we conducted a cognitive task analysis of two specific training competencies. The methodology used in the analysis will form the basis of the query features of the TARGET tool, to be developed in Phase II.

2) Clearly specify and define which features of the game are expected to impact the targeted learning outcomes. The taxonomy developed in Task 2, aids in this endeavor by providing a systematic way to

clearly specify, define, and categorize the specific elements of a game. This, in turn, allows us to select, test, and manipulate specific game features in order to draw useful conclusions about what factors of a game contribute to improved learning outcomes.

3) Develop a theoretically and empirically-based framework that links game characteristics and training objectives. The preliminary crosswalk, developed in Task 3, is a step in that direction, and will be further developed and refined as we gather additional information from planned research studies and from feedback and input from experts in the serious game field.

4) Test the premises of the crosswalk by conducting a well-designed, controlled study that compares the effectiveness of game-based training and conventional training. Specifically:

a) The games used in the study should have clearly defined game elements. The elements should be experimentally manipulated in order to determine their effect on learning outcomes.

b) The study should be conducted in a controlled environment, with clear controls and a variety of sensitive measures. Our testbed training environment at ASU, fulfills this requirement. First of all, at ASU, we have a controlled environment in which methodologically “clean” comparisons between GBT and traditional instruction may be made. Rather than the “nothing” controls that are often used, our control conditions will involve actual instructional alternatives. Second, it is often difficult to track students who have received some form of unique instruction to determine if transfer of training is long-lasting. Because AMT students both stay for four years, and return to ASU in multiple capacities, we will be able to ascertain Level III (Kirkpatrick) transfer effects stemming from GBT manipulations. Third, our targeted subject population is not the ubiquitous undergraduates who characterize much of psychological research. Rather, AMT students are already becoming professionals and, as such, have much in common with our eventual Navy target population. Finally, through our close working relationship with ASU-AMT, we will be able to introduce GBT elements into actual training (vs. special, “one-off” demonstrations) courses and laboratory sessions so that a rich array of training environments will form part of our experimental milieux.

As a final note, we emphasize the importance of designing an instructional game around a good theory of learning – one supported by current research in cognitive science. A consistent finding from cognitive research is that ***one size does not fit all*** – that is, training programs that attempt to teach general thinking skills often end up having limited application. It is when instruction is embedded in a real-world context, where students engage in solving problems similar to the tasks they will be expected to perform later (Mayer, 2003), that meaningful understanding and transfer is more likely to occur.

Chapter 11: Summary of the Proposed Phase II Work

Figure 18 depicts the products we intend to develop in Phases 2-3. During the first year of Phase 2, we will develop the data fields, information content, and software infrastructure for our relational database, TARGET. It will be ‘unpopulated’ until we have collected research data from our primary empirical study, our smaller off-shoot studies, and application of our Design Pattern methodology. These links are shown as colored dots leading to a “populated” tool that will eventually (in Phase 3) become licensable in either its data-rich/design-pattern-rich form or in some scaled down, partially-populated version. Another Year 1 product will be our functional FMS game, including both the main game and the embedded (Pilot’s Arcade) gamelets that are described in the Phase 1 final report. This functional game, besides being a product in its own right, will become the “treatment” variable that is assessed in our GBT impact study that we will conduct at ASU-AMT using its advanced undergraduates as students. We will use an “ungamed” version (i.e., with the gaming elements stripped away but retaining the same instructional material) for our control group. Comparison of the two groups in subsequent venues where FMS programming expertise is needed (simulator training, special FMS practical exercises) will yield high-quality, scientific data concerning the impact of serious game experience upon subsequent training effectiveness. The results of this study will be presented at scientific conferences (e.g., I/ITSEC, Serious Games Summit) and will be submitted to publication in refereed journals (e.g., International Journal of Cognitive Ergonomics).

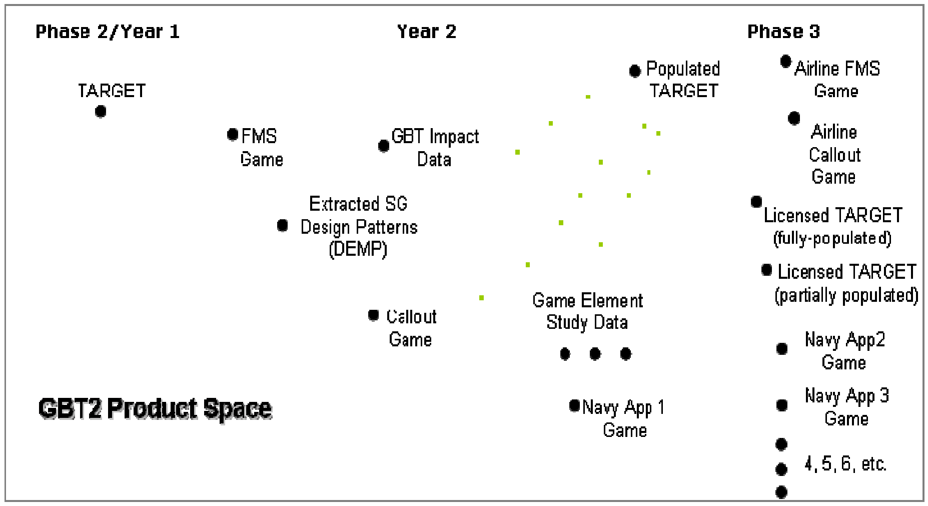


Figure 18. Phase II to III emergence and extension of *GBT2* products.

Later in Phase 2, we will develop a second serious game to promote training in callout-procedures. Like the FMS game, its specifications will be guided by output from the TARGET tool and, importantly, its design will be facilitated by use of the serious-game design patterns using Anacapa's *DPEM* methodology. ROI calculations of the benefits of this method will be computed based on comparisons of time and cost savings in creating the second (callout) game vs. the initial FMS game. Both the FMS and callout games will be subjected to a *DPEM* analysis to provide the basis for selling or licensing these games to the airlines, beginning with Mesa Airlines as the "first-adopter." Our business model for licensing/selling both the individual games and the TARGET tool are described in Section 3 (Technology Insertion) of the transition plan. Another product in Year 2 will be the data coming from several smaller-scale studies (conducted both at Anacapa and at ASU-Mesa) that will examine the impact of individual game elements (or small combinations of elements) on training performance. These will be used to more fully populate TARGET with high-quality research findings.

Starting in Year 1, where customer requirements and specifications are collected, we will begin transitioning our *GBT2* to specific Navy applications. Using a combination of TARGET (to identify the key game element combinations for the targeted training environment) and *DPEM* (to identify the presentation formats of the user interface), we plan to develop and test at least one Navy-specific application in Phase 2. We presently plan to tackle creating a serious game for the navigator of the LCAC (Landing Craft, Air Cushioned), where claimants are ready to receive and test the technology. This crew position has requirements for skill improvement in technical equipment usage (~FMS) and crew coordination (~callouts), so its compatibility with our existing FMS and callout games seems high.

If resources permit, we will explore development and at least partial testing of a second Navy application, with either the submarine (e.g., Approach Officer support displays) or surface ship (e.g., passive sensing) communities serving as suitable targets. We have multiple avenues for adding these communities to the select group of "early adopters", who will succeed our "initial adopter". Our target goal in scoping the content and functionality for the games is that we not exceed $50,000 (including specifications, development, and internal testing) for any one game. Through use of TARGET and *DPEM*, we fully expect the unit costs to decline with repeated transitions to new application areas. These areas will be the focus of an intensive marketing campaign in Phase 3, where groundwork for this expansion will be laid in Phase 2.

References

Aldrich, C. (2005). Learning by doing: A comprehensive guide to simulations, computer games, and pedagogy in e-learning and other educational experiences. Pfeiffer: San Francisco, CA

Allen, S. (2003, December). Formative evaluation. In G. Burgess and H. Burgess (Eds), Beyond intractability. Conflict Research Consortium, University of Colorado, Boulder. http://www.beyondintractability.org/essay/formative_evaluation. Posted December 2003.

Anderson, J. R. (1993) Rules of the mind. Hillsdale, NJ: Erlbaum

Aviation Safety Reporting System (ASRS) website (http://asrs.arc.nasa.gov/)

Bell, H.H. & Waag, W.L. (1998). Evaluating the effectiveness of flight simulators for training combat skills: A review. *International Journal of Aviation Psychology*, *8* (3), 223-242.

Bransford, J., Stevens, R., Schwartz, D., Meltzoff, A., Pea, R., Roschelle, J., Vye, N., Kuhl, P., Bell, P., Barron, B., Reeves, B., & Sabelli, N. (2006). Learning theories and education: Toward a decade of synergy. In P.A. Alexander and P.H. Winne (Eds), *Handbook of Educational Psychology*. Mahwah, NJ: LEA.

Cameron, B. & Dwyer, F. (2005). The effect of online gaming, cognition and feedback typ in facilitating delayed achievement of different learning objectives. *Journal of Interactive Learning Research, 16,* 243-258

Chipman, S.F., Schraagen, J.M., & Shalin, V.L. (2000). Introduction to cognitive task analysis. In J.M. Schraagen, S.F. Chipman, and V.L. Shalin (Eds), *Cognitive task analysis*. Mahway, NJ: LEA.

Conkey, C., Smith, B., DuBuc, C., & Smith, P. A. (2006) Integrating Simulations into Sharable Content Object Reference Model Learning Environments. Paper presented at The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)

Crane, P. & Guckenberger, D. (2000). Above real-time training. In H.F. O'Neil and D.H. Andrews (Eds.), *Aircrew training and assessment*. Mahwah, NJ: LEA.

Desurvire, H., Caplan, M., Toth, J. (2004). Using Heuristics to Improve the Playability of Games. CHI Conference, 2004, Vienna Austria (In the collection of abstracts).

Dick, R.A. & Spiker, V.A. (2007, June). *A pattern framework for designing extreme user environments*. Presented at the Society for Engineering and Research Practices, Las Vegas, NV.

Fisher, J. E. (1976). Competition and gaming: An experimental study. Simulation & Games, 7(3), 321-328.

Fisk, A.D. & Eggemeier, F.T. (1988). Application of automatic/controlled processing theory to training tactical command and control skills: 1. Background and task analytic methodology. In *Proceedings of the Human Factors Society,* 32nd annual meeting (pp 1227-1231). Santa Monica, CA: Human Factors Society.

Fletcher, J. D. & Tobias, S. (2006). Using computer games and simulations for Instruction: A research review. Paper in Proceedings of the Society for Applied Learning Technology Meeting, New Learning Technologies, Orlando, FL, February 2006.

Flight Deck Automation Issues database (www.flightdeckautomation.com).

Garris, R.. & Ahlers. R.A game-based training model: Development, application, and evaluation.. Paper presented at I/ITSEC, November, 2001

Garris, R., Ahlers. R., & Driskell, J. E. (2002). Games, motivation, and learning: A research and practice model. *Simulation and Gaming*, *33*, 441-467.

Gee, J. P. (2005). What would a state of the art instructional video game look like? Innovate August, volume 1, issue 6. http://www.innovateonline.info/index.php?view=article&id=80 retrieved October 2006

Green, C. S. & Bavilier, D. (2003). Action video game modifies visual selective attention. *Nature, 423,* 5347-537.

Gopher, D., Weil, M. & Bareket, T. (1994). Transfer of skill from a computer game trainer to flight. Human Factors, 36, 387-405.

Hart, S. G. & Batiste, V. (1992). Field test of video game trainer. Procedings of the Human Factors Society 36th Annual meeting

Hays, R. T. (2005, November) The *effectiveness of instructional games: A literature review and discussion*. Technical Report. Naval Air Warfare Center: Orlando, FL.

Kintsch, W. (1980). Learning from text, levels of comprehension, or: Why anyone would read a story anyway. *Poetics, 9,* 87-89.

Kirkpatrick, D. L. (1959). Techniques for evaluating training programs. *Journal of the American Society of Training Directors, 13*, 3-9, 21-26.

Kirkpatrick, D. L. (1960). Techniques for evaluating training programs. *Journal of the American Society of Training Directors, 14*, 13-18, 28-32.

Kirkpatrick, D.L. (1987). Evaluation. In R.L. Craig (Ed.), *Training and development handbook: A guide to human resource development* (3rd ed). New York: McGraw-Hill.

Kirkpatrick, D.L. (1996). Great ideas revised. *Training & Development*, January, 54-59.

Lee, K. K., Sanford, B. D., Slattery, R. A. (1997) The human factors of FMS usage in the terminal area. *American Institute of Aeronautics and Astronautics.*

Leemkuil, H., de Jong, T., & Ootes, S. (2000). *Review of educational use of games and simulations*. Project Report Knowledge management Interactive Training System consortium: Ae Enschede, The Netherlands

Lewis, E. & Barlow, M. (2005). The use of games to investigate tactical decision-making. Paper presented at SimTecT05, Sydney, Australia, May 10-12, 2005. Retrieved on September 13, 2006 from www.siaa.asn.au/get/2411853244.pdf

Loukopoulos, L.D., Dismukes, R.K., & Barshi, I. (2003). Cockpit interruptions and distractions: A line observation study. *Proceedings of the 11th International Symposium on Aviation Psychology*, Columbus, OH.

Malone, T. W. (1981) Heuristics for designing enjoyable user interfaces: Lessons from computer games. *Association fro Computing Machinery*

Mayer, R. E. (2003). Learning and instruction. Upper Saddle River, NJ: Prentice Hall.

Mayer, R. E. (2005) Principles for managing essential processing in multimedia learning: Segmenting, pretraining, and modality principles. In R. E. Mayer (Ed.) *The Cambridge handbook of multimedia learning* (pp. 169-182). New York: Cambridge University Press

Mayer, R. E., Mautone, P., & Prothero, W. (2002). Pictorial aids for learning by doing in a multimedia geology simulation game. Journal of Educational Psychology, 94(1), 171- 185.

Mautone, T., Spiker, V.A., & Dick, R.A. (2007, June). *Design Patterns for Serious Games in Tactical Problem Solving* . Presented at the Society for Engineering and Research Practices, Las Vegas, NV.

Marrenbach, J. & Kraiss, K-F. (2000, September). *Advanced flight management system: A new design and evaluation results*. Paper presented at the International Conference on Human-Computer Interaction in Aeronautics.

Miller, C. S., Lehman, J. F., & Koedinger, K. R. (1999). Goals and learning in microworlds. Cognitive Science, 23(3), 305-336.

Morris, C. S., Hancock, P. A., & Shirkey, E. C. (2004). Motivational effects of adding context relevant stress in PC-based game training. *Military Psychology, 16*, 135-147.

Nullmeyer, R.T. & Spiker, V.A. (2003). The importance of crew resource management behaviors in MC-130P mission performance: Implications for training evaluation. *Military Psychology, 15*(1), 77-96*.*

O'Neil, H.F., Wainess, R. & Baker, E. L. (2005). Classification of learning outcomes: Evidence from the computer games literature. *The Curriculum Journal, 16,* 455-474.

Rockwell Collins. (1999, December). *Collins FMS-4200 Flight management system: Pilot's guide*. (2nd ed). Cedar Rapids, IA: Rockwell Collins.

Salden, R. J. C. M., Paas, F. van der Pal, J.,van Merrienboer, J. J. G., (2006) Dynamic task selection in flight management system training. *International Journal of Aviation Psychology, 16, 157-174*

Schunk, D. H. (1991) Self-efficacy and academic motivation. *Educational Psychologist, 26,* 207-231

Scriven, M. (1991). Beyond formative and summative evaluation. In M.W. McLaughlin and E.C. Phillips (Eds), *Evaluation* and education: A quarter century. Chicago: University of Chicago Press.

Shebilske, W. L. , Regian, J. W., Arthur, W., Jordan, J. A. (1992). A dyadic protocol for training complex skills. Human Factors, 1992, 34, 369-374.

Spiker, V.A. & Dick, R.A. (2007, June) Cognitive Design Patterns for Extreme Tactical Environments Presented at the Society for Engineering and Research Practices, Las Vegas, NV.

Sweller, J. (1999) Instructional design in technical area. Camberwell, Australia. ACER Press

Wade, S. E. (1992). How interest affects learning from text. In K.A. Renninger, S. Hidi, & A. Krapp (Eds.), *The role of interest in learning and development* (pp. 255-278). Hillsdale, NJ:Erlbaum.

White, B. Y. & Fredericksen, J. R. (1998) Inquiry, modeling, and Metacognition: Making science assessable to all students. *Cognition and Instruction, 16,* 3-118

Whitehill, B.V. & McDonald, B. A. (1993). "Improving Learning Persistence of Military Personnel by Enhancing Motivation in a Technical Training Program." *Simulation and Gaming, 24*, 294-313

Appendix A: Glossary of Acronyms

ADF	automatic direction finding
AMT	Aeronautical Management Training
APU	auxiliary power unit
ASU	Arizona State University
ATC	air traffic control
CDU	control display unit
CM	crew member
CRJ	Canadair Regional Jet
CTA	cognitive task analysis
DME	distance measuring equipment
EGPWS	early ground proximity warning system
ERJ	Embraaer Regional Jet
FAF	final approach fix
FMS	flight management system
FTD	flight training device
GBT	game based training
GUI	graphical user interface
IAF	initial approach fix
ILS	instrument landing system
INS	inertial navigation system
MFD	multi-function display
OTW	out the window
PBC	profile-based callout
PF	pilot flying
PNF	pilot not flying
SID	standard instrument departure
SME	subject matter expert
STAR	standard terminal approach route
TARGET	Tool for Applying Robust Gaming Elements to Training
VOR	very high frequency omni-directional ranging

Appendix B

Game Based Training (GBT) Formative Evaluation

FMS Game Concept Survey

Arizona State University

Aeronautical Management Training Department

February 2007

Name ____________________________

Title ____________________________

Date ____________________________

Project Overview

Anacapa Sciences and the Department of Aeronautical Management Technology, Arizona State University (ASU-AMT) are working collaboratively under a contract (N0014-06-M-0241) with the Office of Naval Research (ONR) to explore the use of serious games[4] to improve training in aviation skills. Based on work done thus far, we have determined that learning how to program the Flight Management System (FMS) would be a good test case for how one could incorporate a serious gaming approach into student training.

You should have spent at least 20-30 minutes going through the PowerPoint slides that we have created as a prototype demonstration for our FMS serious game concept. We also have created a non-game tutorial exercise for use as a comparison. Once you have that requisite experience with our game prototype, we'd like you to complete the following four-page survey. As with all our research endeavors, your identity will be kept strictly confidential and is used for record-keeping only. No names will ever be attached to the data we report.

The evaluation responses you provide will be vital to us in determining if our FMS game concept has any merit, and if so, what forms the serious game might take. We're hopeful that by incorporating an FMS game into the AMT curriculum, students will be better prepared to operate the FMS during Flight Training Device (FTD) lessons, transition training in the Mesa Air Level D simulator, and importantly, transition to CRJ-series aircraft as a Mesa Air line pilot.

We very much appreciate your time in evaluating our proposed concepts. If you have any questions about this research or would like additional information on the ONR-funded project, please contact one of the following individuals:

Tricia Mautone, PhD	Ron Karp, PhD	Curtis Conkey, PhD
Principal Investigator	Associate Chair	Technical Point of Contact
Anacapa Sciences	Arizona State University Dept of Aeronautical Management Technology	Naval Air Command
301 E. Carrillo St.	7442 E. Tillman Ave	12350 Research Parkway
Santa Barbara, CA 93111	Mesa, AZ 85212	Orlando, Florida 32826-3275
tmautone@anacapasciences.com	ron.karp@asu.edu	Curtis.Conkey@navy.mil
805-966-6157 x13	480-727-1873	(407) 381-8682

[4] Serious games are computer- or video-based games intended to promote training or learning for students who form an audience outside of primary or secondary education.

FMS Game - Features and Functions

1) How would you rate the technical accuracy of the FMS game?

□ **very high** – very accurate with no notable differences	□ **high** – accurate with few notable differences	□ **okay -** fairly accurate with some differences	□ **poor –** low accuracy with many inconsistencies	□ **very poor** – not at all accurate, few consistencies noted
□ not enough information to answer				

On what basis are you making your rating? ____

____

2) How would you rate the game for giving the user a "feel" for the FMS?

□ **very high** – gives a great feel for FMS logic and flow	□ **high** – gives a good feel for FMS logic and flow	□ **reasonable** – user gets some feel for FMS logic and flow	□ **poor** – some key aspects of FMS logic and flow are left out	□ **very poor** – little resemblance to FMS logic and flow
□ not enough information to answer				

On what basis are you making your rating? ____

____

3) How would you rate the game's ability to cover the important functionality of the FMS?

□ **excellent** – covers most of the important FMS functions	□ **good** – covers many important FMS functions	□ **okay** – some important FMS functions are covered	□ **poor** – many key FMS functions are left out	□ **very poor** – most key FMS functions are not covered
□ not enough information to answer				

On what basis are you making your rating? ____

____

4) How would you rate the game's potential to facilitate student FMS training?

□ **very high** – should definitely help student FMS training	□ **high** – it should help student FMS training	□ **mixed** – may or may not help some aspects of student FMS training	□ **poor** – will do little to help student FMS training	□ **very poor** – will do nothing to help student FMS training
□ not enough information to answer				

On what basis are you making your rating? ____

____

5) How would you rate the game's fit with the ASU-AMT training philosophy?

□ **very good** – excellent fit to AMT training philosophy	□ **good** – good fit to AMT training philosophy	□ **okay** – fits some aspects of AMT training philosophy	□ **poor** – will not mesh well with AMT training philosophy	□ **very poor** –at odds with AMT training philosophy
□ not enough information to answer				

On what basis are you making your rating? ____

____

FMS Game – Characteristics of a Game

6) How would you rate the game's potential to maintain the student-user's interest level?

□ **very high** – all students should stay interested	□ **high** – most students should stay interested	□ **reasonable** – some students should stay interested	□ **poor** – few if any students should stay interested	□ **very poor** – no student is likely to stay interested
□ not enough information to answer				

On what basis are you making your rating? ____

____

7) How would you rate the game's potential to be fun and engaging for student pilots?

□ **very high** – a lot of fun, should keep them engaged	□ **high** – should be fun and keep them mostly engaged	□ **reasonable** – could be fun and engaging, but needs work	□ **poor** – not fun or engaging as is	□ **very poor** – not at all fun or engaging
□ not enough information to answer				

On what basis are you making your rating? ____

____

8) Does the game have the features needed to be a successful game like others you have played?

□ **absolutely** – has most of the features needed to be a winner	□ **mostly** – has many of the features needed to be successful	□ **perhaps** – has some features but is missing others	□ **not really** – is missing many essential features	□ **not at all** – has few features found in successful games
□ not enough information to answer				

On what basis are you making your rating?

What features did you like? Which would you add?

9) Would you recommend the game to other student-pilots?

□ **absolutely yes** – I'd recommend it to all students	□ **yes** – I'd recommend it to most students	□ **perhaps** – I might recommend it but with some reservations	□ **probably not** – It's doubtful that I'd recommend it to other students	□ **definitely not** – I wouldn't recommend it to any students
□ not enough information to answer				

On what basis are you making your rating? ____

10) Would you want to use a similar gaming approach to training callouts and profiles?

□ **absolutely yes** – this approach should work well with callouts and profiles	□ **yes** – this approach has potential to work with callouts and profiles	□ **maybe** – the basic approach will need changing to make it work	□ **probably not** – doubtful this approach will work with callouts and profiles	□ **definitely not** – would not use this approach at all for callouts and profiles
□ not enough information to answer				

On what basis are you making your rating? ____

FMS Game – Feature Ratings

11) Rate the following features on their importance for inclusion in an FMS game for student pilots

Interactivity	□ **very important** – should definitely be included	□ **important** – should be included	□ **perhaps** – maybe include in some form	□ **not important** – shouldn't be included	□ **definitely not important** – shouldn't even consider
Immediate Feedback	□ **very important** – should definitely be included	□ **important** – should be included	□ **perhaps** – maybe include in some form	□ **not important** – shouldn't be included	□ **definitely not important** – shouldn't even consider
Challenge	□ **very important** – should definitely be included	□ **important** – should be included	□ **perhaps** – maybe include in some form	□ **not important** – shouldn't be included	□ **definitely not important** – shouldn't even consider
Time Pressure	□ **very important** – should definitely be included	□ **important** – should be included	□ **perhaps** – maybe include in some form	□ **not important** – shouldn't be included	□ **definitely not important** – shouldn't even consider
Command-sounding Voice-overs	□ **very important** – should definitely be included	□ **important** – should be included	□ **perhaps** – maybe include in some form	□ **not important** – shouldn't be included	□ **definitely not important** – shouldn't even consider
Publicized Scores	□ **very important** – should definitely be included	□ **important** – should be included	□ **perhaps** – maybe include in some form	□ **not important** – shouldn't be included	□ **definitely not important** – shouldn't even consider
Realistic Graphics	□ **very important** – should definitely be included	□ **important** – should be included	□ **perhaps** – maybe include in some form	□ **not important** – shouldn't be included	□ **definitely not important** – shouldn't even consider
Animation	□ **very important** – should definitely be included	□ **important** – should be included	□ **perhaps** – maybe include in some form	□ **not important** – shouldn't be included	□ **definitely not important** – shouldn't even consider
Sound	□ **very important** – should definitely be included	□ **important** – should be included	□ **perhaps** – maybe include in some form	□ **not important** – shouldn't be included	□ **definitely not important** – shouldn't even consider
Other ________________	□ **very important**	□ **important**	□ **perhaps**	□ **not important**	□ **definitely not important**
Other ________________	□ **very important**	□ **important**	□ **perhaps**	□ **not important**	□ **definitely not important**

12) Which features did you like best?

13) Was there anything confusing or distracting about any of the game features?

14) Was the interaction between the virtual characters (i.e., the Captain, the ATC, etc.) and the player realistic?

15) Please note any other comments that you have concerning the FMS game concept you have reviewed: